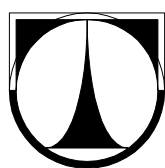


**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky a mezioborových inženýrských studií



# **DIPLOMOVÁ PRÁCE**

**2007**

**Libor Mušák**

# **TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M 2612 – Elektrotechnika a informatika

Studijní obor: 3906T001 – Mechatronika

## **Automatizované měření izolačního odporu izolací v elektrických strojích**

## **Automatic measurement of insulating resistance of slot insulation in electric devices**

### **Diplomová práce**

Autor: **Libor Mušák**

Vedoucí práce: Ing. Leoš Beran

Konzultant: Ing. Tomáš Mikolanda

# TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra elektrotechniky

Akademický rok: 2006/2007

## ZADÁNÍ DIPLOMOVÉ PRÁCE

pro: Libora Mušáka

studijní program: M 2612 – Elektrotechnika a informatika

obor: 3906T001 - Mechatronika

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu: **Automatizované měření izolačního odporu izolací v el. strojích**

Zásady pro vypracování:

1. Seznamte se s možnostmi automatizovaného měření a zvolte vhodné řešení pro měření izolačního odporu izolací v el. strojích.
2. Vytvořte ovládací program pro PC, který umožní obecné nastavení měření a grafické zpracování výsledků.
3. Ovládací program pro PC naprogramujte modulárně tak, aby bylo možné jeho pozdější rozšíření. K programu vytvořte přehledný komentář.
4. K navrženému programu vytvořte nápovědu pro pozdější použití.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40 až 50 stran

Seznam odborné literatury:

[1] BACH P. – URBÁNEK S. *Automatizace měření*. Praha: České vysoké učení technické, 1993.

[2] HAAZ V. – ROZTOČIL J. – NOVÁK J. *Číslicové měřicí systémy*. Vydání druhé přepracované. Praha: Ediční středisko ČVUT, 2000. 315 s. ISBN 80-01-02219-6.

[3] ĎAĎO S. – KREIDL M. *Senzory a měřicí obvody*. Vydání první. Praha: Ediční středisko ČVUT, 1996. 315 s. ISBN 80-01-01500-9.

Vedoucí diplomové práce: Ing. Leoš Beran

Konzultant: Ing. Tomáš Mikolanda

Zadání diplomové práce: **16.10.2006**

Termín odevzdání diplomové práce: **18. 5. 2007**

L.S.

.....  
Vedoucí katedry

.....  
Děkan

V Liberci dne 16.10.2006

## Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce, a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum:

Podpis:

## **Poděkování**

Na tomto místě bych rád poděkoval všem lidem, kteří mají jakoukoli zásluhu na realizaci této práce. Jmenovitě bych chtěl poděkovat panu Ing. Leošovi Beranovi, vedoucímu mé diplomové práce, za neocenitelné rady, poznámky a trpělivost při konzultacích. Mé poděkování patří i konzultantovi, panu Ing. Tomášovi Mikolandovi, za odborné poznámky při programování vlastní aplikace. Na závěr bych chtěl poděkovat rodině, svým kamarádům a přítelkyni Šárce za podporu při studiích a tvorbě této práce.

## **Abstrakt**

Cílem této diplomové práce bylo navrhnout vhodné řešení pro měření izolačního odporu drážkové izolace. Realizované pracoviště se skládá z řídicího počítače, ampérmetru a zdroje napětí. Tyto přístroje mezi sebou komunikují po sběrnici GPIB.

Součástí tohoto návrhu byla tvorba aplikace, která umožní ovládat přístroje nezbytné pro měření. Vlastní aplikace umožňuje uživateli připojení/odpojení obou přístrojů a jejich identifikaci. Dále nastavení parametrů měření, samotné měření a vyhodnocení naměřených dat. Vyhodnocení dat je realizováno pomocí statistických výpočtů a tato výsledná data lze uložit do souboru nebo vykreslit do grafu. Aplikace byla naprogramována v prostředí BUILDER C++.

Aplikace obsahuje nápovědu, protože se předpokládá používání této aplikace dalšími uživateli. Nápověda byla vytvořena pomocí programu Microsoft Help Workshop a obsahuje témata pro všechna nastavení a funkce aplikace. Vytvořená aplikace byla podrobena důkladnému testování při reálném měření, které potvrdilo úspěšný návrh celé aplikace i její nápovědy.

Klíčová slova: automatizované měření, měření proudu, izolační odpor, GPIB

## **Abstract**

The goal of my diploma thesis was to design a suitable solution for automatic measurement of insulating resistance of slot insulation. The workplace is realized by a computer, an ampere-meter and voltage source. These devices communicate through GPIB bus.

A part of solution was an application for measuring devices control. The main application is able to connect/disconnect devices and their identification. The application allows setting of measurement options, does measuring and evaluates results. Evaluation of results is realized by statistical calculations and then we can save these results in a data file or draw them as a chart. The application was realized in BUILDER C++.

The application contains a help, because it is supposed, that this application will be used by more users. The help was created with the aid of Microsoft Help Workshop and includes topics for all parameters and functions of the application. This application

was tested during a real measuring. This measuring verified rightness of the whole application and its help.

Keywords: automatic measurement, current measurement, insulating resistance, GPIB



## Obsah

<b>ABSTRAKT .....</b>	<b>5</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>9</b>
<b>SEZNAM SYMBOLŮ .....</b>	<b>10</b>
<b>SEZNAM ZKRATEK .....</b>	<b>10</b>
<b>SLOVNÍK POJMŮ .....</b>	<b>11</b>
<b>ÚVOD.....</b>	<b>12</b>
<b>1 MĚŘENÍ IZOLAČNÍHO ODPORU .....</b>	<b>13</b>
<b>2 MĚŘÍCÍ SYSTÉMY .....</b>	<b>15</b>
2.1 Rozhraní GPIB .....	16
2.1.1 Základní parametry .....	17
2.1.2 Interfaceové zprávy .....	18
2.1.3 Přístrojové zprávy .....	18
2.1.4 Možnosti spojení přístroje s počítačem .....	18
2.2 Programování měřícího systému s rozhraním GPIB .....	20
2.2.1 SCPI (standard commands for programmable instruments).....	21
2.2.2 Vlastní programování přístrojů pomocí knihovny VISA .....	22
<b>3 MĚŘÍCÍ APARATURA.....</b>	<b>25</b>
<b>4 REALIZACE APLIKACE.....</b>	<b>27</b>
4.1 Funkce aplikace .....	27
4.2 Třída Přístroj .....	29
4.2.1 Třída Ampermetr .....	32
4.2.2 Třída Zdroj .....	33
4.3 Vzhled a funkce aplikace .....	34
4.3.1 Záložka Připojení .....	35
4.3.2 Záložka Nastavení.....	36
4.3.3 Záložka Měření .....	38
4.3.4 Záložka Vyhodnocení .....	39
4.3.5 Záložka Graf .....	42
4.4 Popis užitých komponent.....	43
4.5 Popis vybraných funkcí .....	44
<b>5 TVORBA NÁPOVĚDY .....</b>	<b>49</b>
5.1 Software pro tvorbu nápovědy.....	49

---

5.1.1	WinHelp.....	49
5.1.2	HTML Help .....	51
5.2	Realizace nápovědy .....	51
5.2.1	Témata nápovědy .....	52
5.2.2	Obsah a rejstřík nápovědy.....	53
<b>6</b>	<b>OVĚŘOVACÍ MĚŘENÍ.....</b>	<b>55</b>
6.1	Vyhodnocení výsledků .....	55
6.2	Výsledky ověřovacího měření .....	57
	<b>ZÁVĚR .....</b>	<b>61</b>
	<b>LITERATURA.....</b>	<b>63</b>
	<b>PŘÍLOHA A FOTODOKUMENTACE.....</b>	<b>64</b>
	<b>PŘÍLOHA B TABULKY NAMĚŘENÝCH HODNOT .....</b>	<b>65</b>
	<b>PŘÍLOHA C TABULKA POUŽITÝCH PŘÍSTROJŮ.....</b>	<b>68</b>
	<b>PŘÍLOHA D OBSAH CD.....</b>	<b>69</b>

## Seznam obrázků

Obrázek 1.1 – Schéma zapojení.....	13
Obrázek 1.2 – Měřicí aparatura pro manuální měření.....	14
Obrázek 2.1 – Struktura zapojení RS–232 a), struktura zapojení GPIB b) [3].....	15
Obrázek 2.2 – Topologie zapojení přístrojů pomocí GPIB .....	16
Obrázek 2.3 – Konektory GPIB kabelu [4] .....	19
Obrázek 2.4 – Zásuvná karta do PC s PCI/GPIB rozhraním [5] .....	19
Obrázek 2.5 – Zásuvná karta do notebooku s PCMCIA/GPIB rozhraním [6] .....	19
Obrázek 2.6 – Kabel s rozhraním USB/GPIB [5].....	20
Obrázek 3.1 – Zapojení měřicí aparatury .....	25
Obrázek 4.1 – Vývojový diagram aplikace .....	28
Obrázek 4.2 – Záložka Připojení .....	35
Obrázek 4.3 – Záložka Nastavení .....	36
Obrázek 4.4 – Kolonka kolikrát změřit jeden vzorek .....	37
Obrázek 4.5 – Záložka Měření .....	38
Obrázek 4.6 – Záložka Vyhodnocení .....	41
Obrázek 4.7 – Záložka Graf.....	42
Obrázek 4.8 – Ukázka ošetření uživatelských chyb .....	48
Obrázek 5.1 – Ukázka okna nápovědy WinHelpu.....	50
Obrázek 5.2 – Ukázka okna HTML helpu [11] .....	51
Obrázek 5.3 – Ukázka tématu nápovědy .....	52
Obrázek 5.4 – Struktura obsahu nápovědy .....	53
Obrázek 5.5 – Ukázka struktury témat v kapitolách.....	54
Obrázek 5.6 – Rejstřík nápovědy.....	54
Obrázek 6.1 – Tabulka naměřených hodnot .....	55

## Seznam symbolů

$\bar{I}$	[A]	průměrná hodnota proudu změřených vzorků
$N$	–	počet změřených vzorků
$I_i$	[A]	proud protékající daným vzorkem
$\sigma$	–	směrodatná odchylka
$\Delta \bar{I}$	[A]	chyba průměrné hodnoty
$U_{VYSL}$	[V]	napětí na vzorku
$\Delta U_{VYSL}$	[V]	chyba napětí na vzorku
$U_{ZDR}$	[V]	napětí zdroje
$R$	[M $\Omega$ ]	hodnota ochranného odporu
$\Delta R$	[M $\Omega$ ]	chyba ochranného odporu
$I$	[A]	proud tekoucí obvodem
$\Delta I$	[A]	chyba proudu tekoucího obvodem

## Seznam zkratk

ŘJ	řídící jednotka
FJ	funkční jednotka
HP	Hewlett Packard
GPIB	general purpose interface bus
VISA	virtual instruments systems architecture
SCPI	standard commands for programmable instruments
DLL	dynamic link library
VPP–2	VXI Plug&Play (the system frameworks specification)

**Slovník pojmů**

GPiB	komunikační rozhraní mezi PC a přístroji
VISA	knihovna funkcí pro komunikaci s přístroji (např. po GPiB)
SCPI	standard příkazů pro ovládání přístrojů
Klíčové slovo	specifikuje dané téma v nápovědě
Help	nápověda
Zoom	funkce pro přiblížení průběhů v grafu
Grid	mřížka zobrazovaná v grafu
ComboBox	grafická komponenta umožňující pomocí rolovacího seznamu výběr jedné hodnoty ze seznamu nabízených
Checkbox	grafická komponenta umožňující „zaškrtnutí“ položky
Edit	grafická komponenta umožňující vepsání textu
StatusBar	panel nacházející se v dolní části aplikace
RadioGroup	grafická komponenta umožňující jednu volbu z několika možností
PageControl	grafická komponenta sloužící k vytváření záložek
Session	proměnná představující odkaz na spojení s přístrojem nebo na inicializování systému VISA a získání spojení

## Úvod

Diagnostika asynchronního motoru a jeho jednotlivých částí je v dnešní době velmi důležitá při jeho provozu. Jednou z takových částí motoru je drážková izolace, která je plošná a odděluje vinutí motoru od jeho kostry. Vnější vlivy (např. teplo, vlhko...) mají negativní dopad na životnost této izolace, resp. celého motoru. Proto by jistě byl popis vlivu teplotního zatížení na životnost drážkové izolace pro samotné výrobce velkým přínosem. Při popsání této závislosti by bylo možné zpětné zlepšení materiálů, z nichž je izolace vyrobena, a tím dosáhnout větší životnosti motoru. Prodloužení životnosti motoru by zajisté mělo u uživatelů velmi příznivý ohlas. Toto téma je právě řešeno na Výzkumném ústavu mechatroniky a technické informatiky na Technické univerzitě v Liberci. Má diplomová práce by měla být přínosem pro další výzkum v této oblasti technické diagnostiky.

Tato diplomová práce navazuje na moji práci, kterou jsem započal v ročníkovém projektu **Měření izolačního odporu drážkové izolace**. V projektu bylo navrženo a realizováno pracoviště, na kterém je možné měřit vzorky izolací a vyhodnocovat vliv teploty na jejich izolační odpor. Měření izolačního odporu je velmi časově náročné vzhledem k transienčním dějům, které probíhají v izolačních materiálech. Proto je nutné celé měření zautomatizovat, stejně jako zpracování výsledků.

Hlavním cílem práce je návrh a realizace aplikace, která by umožňovala automatické měření izolačního odporu drážkových izolací používaných v asynchronních motorech. Je nutné zvolit nejvhodnější rozhraní pro komunikaci měřících zařízení s řídicím PC a dále realizovat aplikaci, která bude komunikaci a vlastní měření vykonávat. Aplikace by měla být pro uživatele intuitivní, s co nejsnazší obsluhou. K tomuto účelu je nutné vytvořit také nápovědu, která bude detailně popisovat funkce aplikace a její možné nastavení. Celý zdrojový kód programu bude realizován co nejpřehledněji a nejobecněji, aby byla umožněna snadná a rychlá modifikace celé aplikace.

Automatickým měřením snížíme časovou náročnost kladenou na uživatele, protože měření může přesáhnout i desítky minut. Dojde také ke snížení časové náročnosti při vyhodnocování výsledků tím, že aplikace bude výsledky vyhodnocovat automaticky.

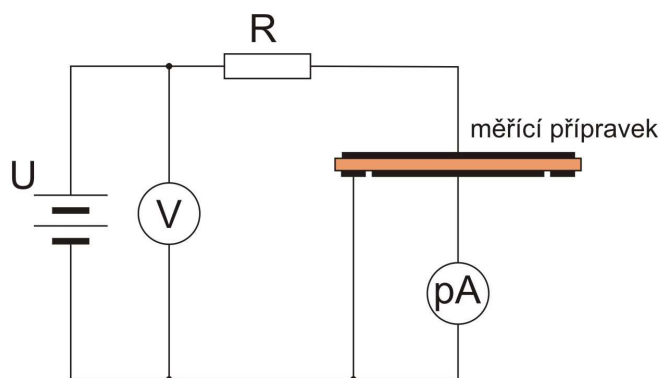
## 1 Měření izolačního odporu

V této kapitole nastíním základní princip, jak úspěšně změřit izolační odpor izolačního materiálu. V zásadě se jedná o měření VA charakteristiky tohoto materiálu a následné vypočítání jeho odporu (podle Ohmova zákona).

Při měření elektrického odporu izolantů je nutné rozlišit vnitřní, povrchový a izolační odpor. Vnitřní odpor je roven poměru stejnosměrného napětí a proudu, který teče vnitřkem vzorku mezi dvěma elektrodami. Povrchový odpor je definován poměrem napětí a proudu, který teče povrchem vzorku. Izolační odpor je určen poměrem napětí a součtem proudů, které tečou povrchem a vnitřkem vzorku [1]. Díky znečištění a vlhkosti, která významně ovlivňuje povrchový odpor izolantu, bereme vnitřní odpor jako relevantní výslednou hodnotu odporu izolantu [2].

Tento odpor (resp. VA charakteristiku) můžeme měřit vhodným zapojením měřící aparatury (Obrázek 1.1). Měřící obvod se tedy skládá ze zdroje napětí, voltmetru, ampérmetru, ochranného odporu a měřícího přípravku.

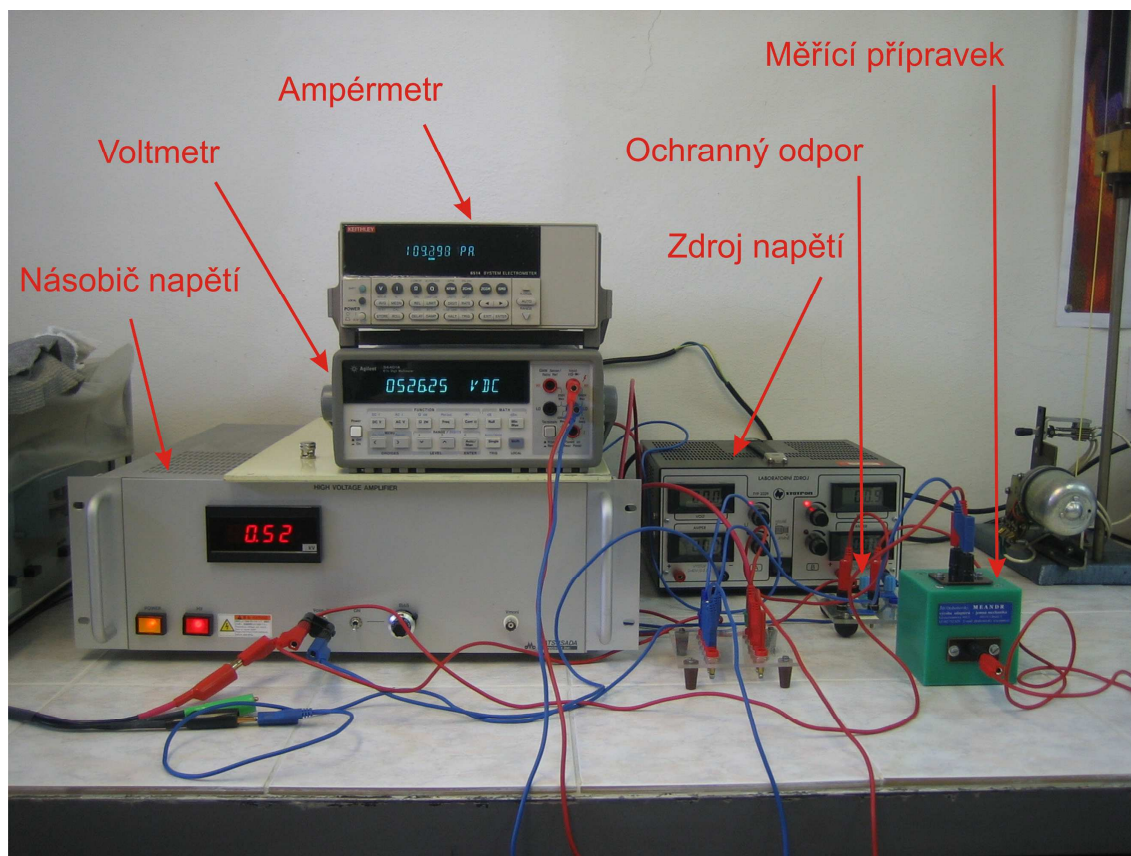
Měřící přípravek je zkonstruován tak, že je vzorek izolace umístěn mezi tři elektrody [2]. V přípravku je umístěna napěťová elektroda, měřící elektroda a ochranný prstenec. Ochranný prstenec má úlohu odvést povrchový proud mimo ampérmetr, aby byl měřen pouze proud procházející vzorkem (vzorek má oranžovou barvu).



Obrázek 1.1 – Schéma zapojení

Pro měření odporu izolačních materiálů je zapotřebí zdroj vysokého napětí (řádově jednotky kV). Pro dosažení tohoto napětí byl použit násobič napětí, který  $500 \times$  znásobil přivedené napětí z manuálně ovládaného zdroje. Ochranný odpor ( $R$ ) musí být v obvodu zapojen z důvodu ochrany ampérmetru. Pokud by měřená izolace byla proražena (ztratila by své izolační vlastnosti), narostlo by na ampérmetru napětí na úroveň napětí zdroje a tím by mohl být ampérmetr poškozen [2].

Podle schématu (Obrázek 1.1) bylo v projektu [2] realizováno pracoviště, které umožňuje měřit VA charakteristiku izolantu. Obrázek 1.2 toto pracoviště zobrazuje.



Obrázek 1.2 – Měřicí aparatura pro manuální měření

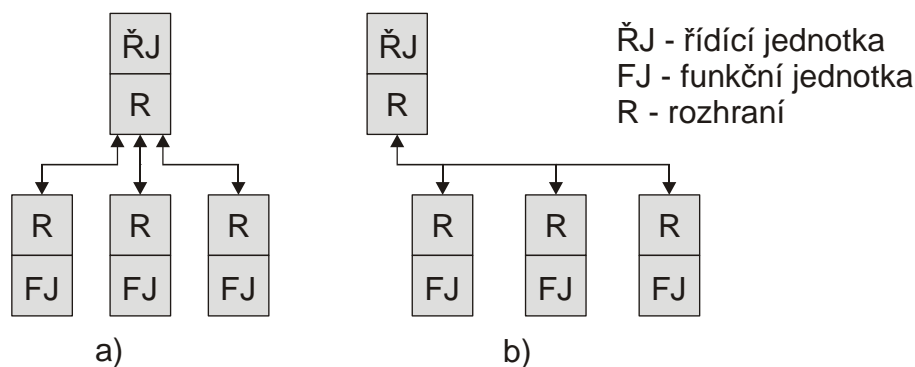
Z důvodu automatizace měření je kladen na přístroje požadavek, aby umožňovali komunikaci s řídicím PC. Z tohoto důvodu není použit zdroj napětí uvedený v literatuře [2], protože neumožňuje komunikaci s PC. Ampérmetr zůstal stejný, protože komunikaci po sběrnici GPIB nebo po sériové lince RS232 umožňuje. Násobič napětí zůstal také stejný. Komunikační rozhraní přístrojů s PC jsem zvolil na základě poznatků uvedených v kapitole 2.



## 2 Měřicí systémy

Za měřicí systém lze považovat takové uspořádání zařízení a jejich součástí (propojení, měřicí přípravek apod.), které je schopno automaticky změřit danou úlohu. Základem měřicího systému je počítač (řídící jednotka, ŘJ), pomocí kterého se ovládají měřicí přístroje (funkční jednotky, FJ). Může tedy jít o sestavu přístrojů schopnou pouze změřit danou veličinu, odeslat ji do počítače, vyhodnotit a zpracovat výsledky. Nebo může jít o systém, kde se jedná o řízení komplexního měření, a kde se nastavují potřebné akční veličiny na základě odměřených hodnot sledovaných veličin. Tento popis měřicích systémů odpovídá popisu V. Haasze, 2000 [3] str. 9.

Dále se budu věnovat nejčastějším způsobům spojení měřicí a funkční jednotky. Pro laboratorní účely je typický systém s jednoduchou rekonfigurovatelností a krátkou komunikační vzdáleností. V řadě případů se klade velký nárok na dynamiku měřicího systému. Nejpoužívanějšími rozhraními jsou RS-232 (sériová linka) a rozhraní IEEE 488 (GPIB). Obrázek 2.1 zobrazuje strukturu obou rozhraní. RS-232 má strukturu hvězdicovou, zatímco GPIB sběrniceovou. Sériové rozhraní je značně omezeno počtem sériových portů v počítači.



Obrázek 2.1 – Struktura zapojení RS-232 a), struktura zapojení GPIB b) [3]

Tato dvě nejpoužívanější rozhraní budou v budoucnu postupně nahrazována sériovým přenosem pomocí rozhraní USB, které má stromovou strukturu a rychlost přenosu srovnatelnou s rozhraním GPIB. Hlavní výhodou USB rozhraní oproti stávajícím je, že není nutné instalovat nový hardware jako v případě rozhraní GPIB a jeho jednoduché připojení k PC.

Parametry rozhraní RS-232 byly již mnohokrát detailně popsány a lze je nalézt např. v publikaci V. Haasze, 2000 [3] str. 34. V této diplomové práci se dále věnuji

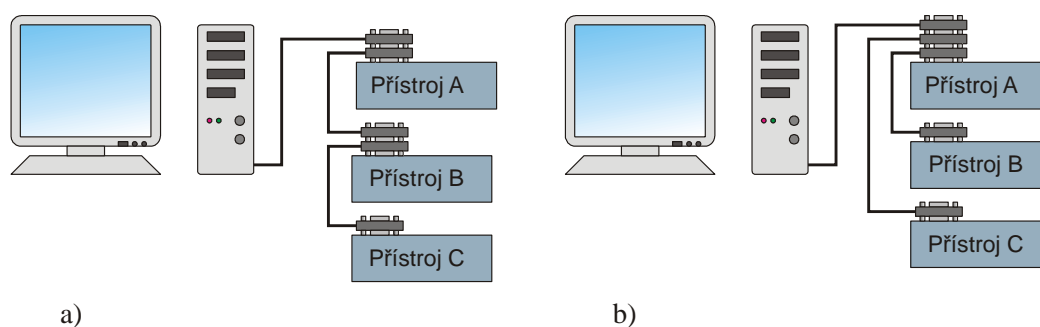
pouze rozhraní GPIB, které je pro realizaci práce využito. Toto rozhraní jsem zvolil, protože má široké využití. Většina špičkových laboratorních přístrojů používá právě rozhraní GPIB ke svému ovládání. Další zajímavou vlastností je relativně jednoduché připojení přístrojů k PC díky novým hardwarovým prostředkům, které jsou popsány v kapitole 2.1.4.

## 2.1 Rozhraní GPIB

V tomto odstavci uvedu pouze základní informace použitého rozhraní GPIB. Podrobnější popis rozhraní lze nalézt na internetu nebo v publikaci V. Haasze, 2000 [3] str. 18, ze které jsem čerpal následující informace.

Tento systém byl navržen firmou HP v roce 1972 a později byl ustaven jako norma IEEE 488, resp. novější norma IEEE 488.1 a IEEE 488.2. Jedná se o systém složený z autonomních přístrojů. To umožňuje sestavovat flexibilní měřicí systémy, přičemž jednotlivé přístroje mohou pracovat zcela autonomně.

Zapojení funkčních jednotek k počítači může mít hvězdicovou strukturu, lineární strukturu, nebo jejich kombinaci. Typy těchto zapojení ilustruje Obrázek 2.2, kde část a) reprezentuje lineární zapojení a část b) hvězdicové zapojení. Připojení funkčních jednotek k počítači se většinou realizuje pomocí zásuvných karet, které zajišťují komunikaci mezi rozhraním počítače (např. USB nebo PCI) a rozhraním měřicího systému (GPIB). Zařízení se k počítači připojuje kabelem s 24 pinovým konektorem (viz Obrázek 2.3).



Obrázek 2.2 – Topologie zapojení přístrojů pomocí GPIB

### 2.1.1 Základní parametry

- Počet funkčních jednotek maximálně 15
- Celková délka sběrnice max. 20 m
- Vzdálenost mezi dvěma funkčními jednotkami max. 2 m
- Počet vodičů sběrnice 24
- Maximální přenosová rychlost 1 MB/s
- Elektrické úrovně signálu TTL, L(<0,8 V), H(>2,0 V)
- Logické úrovně signálu Log1 (True) ~ L, Log0 (False) ~ H

Pro komunikaci slouží 8 datových vodičů (DIO1 - DIO8), po kterých jsou oběma směry přenášena data, adresy funkčních jednotek a vícevodičové signály. Dále je tu 5 vodičů pro vysílání jednovodičových zpráv (ATN, IFC, REN, SRQ, EOI), 3 vodiče pro řízení přenosu dat (DAV, NRFD, NDAC) a 8 vodičů zemnicích. Při komunikaci rozlišujeme dva druhy zpráv (přístrojové a interfaceové zprávy). Tabulka 2.1 uvádí popis jednotlivých pozic signálů rozhraní GPIB. Výhoda sběrnice GPIB je ta, že zaručuje 100% přenos dat.

pin	signál	popis	pin	signál	popis
1	DIO 1	přenos dat	13	DIO 5	přenos dat
2	DIO 2	přenos dat	14	DIO 6	přenos dat
3	DIO 3	přenos dat	15	DIO 7	přenos dat
4	DIO 4	přenos dat	16	DIO 8	přenos dat
5	EOI	end or identify	17	REN	remote enable
6	DAV	data valid	18	GND	zem DAV
7	NRFD	not ready for data	19	GND	zem NRFD
8	NDAC	not data accepted	20	GND	zem NDAC
9	IFC	interface clear	21	GND	zem IFC
10	SRQ	service request	22	GND	zem SRQ
11	ATN	attention	23	GND	zem ATN
12	SCHIELD	zem	24	GND	signálová zem

Tabulka 2.1 – Popis pinů a signálů na sběrnici GPIB

Jednotky měřicího systému se mohou nacházet ve třech stavech: posluchač, mluvčí nebo řidič. Řidič (v našem případě počítač) plně ovládá činnost celého systému a může vysílat zprávy IFC, REN, ATN a EOI. Tato jednotka určuje kdo je posluchač a kdo mluvčí. V konkrétním okamžiku může být pouze jedna jednotka v roli mluvčího, posluchačů může být několik. Také řidič, který vysílá jednovodičové zprávy, může být

v konkrétním okamžiku pouze jeden. Systém může obsahovat více řidičů, ale pouze jeden má charakter systémového kontroléru, který vždy zahajuje řízení systému.

### **2.1.2 Interfaceové zprávy**

Interfaceové zprávy se používají k řízení činnosti funkčních jednotek (přístrojů) prostřednictvím jejich stykových funkcí. Mohou být vysílány pouze řídicí jednotkou. Vícevodičové interfaceové zprávy jsou vysílány po datové sběrnici (DIO1–DIO8) spolu s jednovodičovou zprávou ATN = True (attention). Mezi tyto zprávy patří adresy a příkazy.

Vícevodičové příkazy se dělí na univerzální, adresované a sekundární. Univerzální příkazy vyvolávají odpovídající činnost u všech zařízení připojených k systému bez jejich adresování. Adresované příkazy vyvolají činnost pouze u zařízení, které je adresováno. Mezi sekundární příkazy lze zařadit zrušení nebo umožnění paralelního hlášení.

Jednovodičové příkazy jsou zprávy vysílané pomocí signálů IFC, REN a ATN.

### **2.1.3 Přístrojové zprávy**

Přístrojové zprávy jsou vysílány současně s ATN = False. Jsou vázány na činnost přístrojové části funkčních jednotek. Obvody rozhraní na tyto zprávy nereagují a pouze zabezpečují jejich přenos. Přístrojové zprávy jsou například programovací příkazy (znaky a řetězce), naměřená data a informace o přístroji.

### **2.1.4 Možnosti spojení přístroje s počítačem**

Připojení přístroje k počítači se provádí pomocí zásuvných karet do počítače (rozhraní PCI nebo PCMCIA). Další možností je připojení k USB sběrnici počítače. Vždy je nutné nainstalovat ovladače, které nám zajistí komunikaci s daným hardwarem. Spojení mezi počítačem a vlastním přístrojem je realizováno pomocí kabelů (Obrázek 2.3).



Obrázek 2.3 – Konektory GPIB kabelu [4]

### **Možnosti připojení zařízení s počítačem:**

- 1) Komunikace s počítačem je zajištěna pomocí zásuvné karty do počítače (PCI/GPIB rozhraní), která zprostředkuje komunikaci mezi řídicí jednotkou a funkční jednotkou (Obrázek 2.4 znázorňuje tuto zásuvnou kartu). Nevýhoda těchto typů karet je ta, že při připojení nebo odpojení karty je vždy nutné vypnout počítač. Karta s PCI rozhraním je též komplikovaná na montáž (nutno otevřít PC).



Obrázek 2.4 – Zásuvná karta do PC s PCI/GPIB rozhraním [5]

- 2) Pro notebooky jsou k dispozici podobné zásuvné karty, ale s rozhraním PCMCIA/GPIB (Obrázek 2.5). Nevýhoda je stejná jako u karty PCI/GPIB (nutnost vypnout PC při instalaci karty).



Obrázek 2.5 – Zásuvná karta do notebooku s PCMCIA/GPIB rozhraním [6]

- 3) Další a zároveň nejvýhodnější možností je použití kabelu s rozhraním USB/GPIB (Obrázek 2.6). Toto rozhraní nám umožní rychlé propojení počítače s funkční jednotkou. Zařízení lze pak použít pro jakýkoli počítač vybavený USB rozhraním (v dnešní době téměř u všech počítačů). Největší výhodou tohoto zařízení je to, že odpadá složitá montáž která byla nutná v případě PCI karty. Zařízení lze kdykoli připojit či odpojit bez vypínání počítače.



Obrázek 2.6 – Kabel s rozhraním USB/GPIB [5]

## 2.2 Programování měřicího systému s rozhraním GPIB

Funkční jednotky lze ovládat pomocí řídicí jednotky několika způsoby. Samotné ovládání je založeno na přenosu zpráv mezi řídicí a funkční jednotkou. Nejprve je nutné nainstalovat ovladače k příslušnému hardwaru, který zprostředkovává komunikaci počítače s rozhraním GPIB. Dále následuje sled příkazů, které funkční jednotka vykonává.

Ovládat funkční jednotku lze těmito druhy příkazů:

- SCPI příkazy
- Přístrojové příkazy (jsou obsaženy ve standartu SCPI)

Samotné programování probíhá v některém ze softwarů využívající jazyk C, který se jeví jako optimální, jak popisuje V. Haasz, 2000 [3] str.153. Jsou to např. programy Microsoft Visual C++ a Borland C++ Builder. Také je možné využít programy přímo určené pro vývoj aplikací na ovládání měřicích přístrojů jako je LabWindows/CVI nebo LabVIEW. Tyto programy mají implementovány komponenty přímo určené k měření, jako jsou grafy, různé matematické funkce apod. Jako další lze uvést software MATLAB a jeho grafické rozhraní GUI.

### 2.2.1 SCPI (standard commands for programmable instruments)

SCPI je obecný standard pro ovládání přístrojů od jakéhokoli výrobce. Je to souhrn příkazů pro komunikaci mezi řídicí a funkční jednotkou. Tyto příkazy nejsou závislé na technickém řešení rozhraní ani na použitém protokolu přenosu dat [3]. Tento standard definuje:

#### 1) Množinu SCPI příkazů pro ovládání přístroje

Tyto příkazy nám umožňují plně ovládat jakýkoli přístroj se všemi jeho funkcemi. Někdy je možné definovat i nové příkazy, ale je nutné, aby byly vytvořeny v souladu se standardem SCPI. Tyto příkazy jsou většinou dodávány v manuálu přístroje a mají stromovou strukturu. Zde uvádím ukázkou struktury SCPI příkazů pro přístroj KEITHLEY 6514:

```
:FORMat
  [:DATA] <type>[,<length>]  Specify data format; ASCii,
                              REAL 32 or SREal
  [:DATA]?                    Query data format.
:ELEMents <item list>        Specify data elements;
                              READing, TIME and STATus.
:ELEMents?                    Query data format elements.
:BORDer <name>                Specify byte order; NORMAl or
                              SWAPped
:BORDer?                      Query byte order
```

#### 2) Množinu obecných příkazů

Nejzákladnější příkazy jsou tzv. obecné přístrojové příkazy. Jedná se o příkazy vysílané řídicí jednotkou, které jsou určeny pro ovládání přístrojů na obecné úrovni a pro identifikaci jejich stavu. Příkazy mají tvar \*XXX nebo \*XXX?. XXX je název příkazu a otazník značí to, zda se jedná o příkaz typu dotaz (query), kdy je přístroj povinen řídicí jednotce odeslat odpověď. Jako příklad příkazů uvedu ty, které jsem použil ve své aplikaci.

- \*RST – nulování přístroje (ukončení všech operací)
- \*IDN? – dotaz na specifikaci přístroje

### 3) Formát dat

Formáty dat jsou rozděleny na formáty pro přijímání a vysílání dat. Oba druhy se nepatrně liší. Vysílat lze příkazy v přesném formátu (SCPI příkazy), ale přijímat lze „jakýkoli“ formát.

### 4) Stavový model přístroje

Stavový model přístroje je schéma funkční struktury reálného přístroje dle standardu SCPI.

### 5) Model spouštění přístroje

Tento subsystém je určen k synchronizaci přístroje s vnějšími událostmi. Definuje jak by měl být přístroj naprogramován, aby vykonal sekvenci za sebou jdoucích příkazů (např. jak definovat zdroj spouštění nebo časovou prodlevu mezi měřeními apod.). Přístroj pak čeká na spouštěcí událost. Nastane-li tato událost, přístroj provede sekvenci příkazů.

## 2.2.2 Vlastní programování přístrojů pomocí knihovny VISA

Snahou všech výrobců měřících přístrojů je to, aby docílili vzájemné spolupráce svých přístrojů s přístroji jiného výrobce. Proto vznikly tzv. systémové rámce (definované standardem VPP-2), které definují hardwarovou konfiguraci řídicího počítače, ovladače přístrojů, aplikační vývojové prostředky aj. Největší pozornost je věnována standardizaci ovladačů měřících přístrojů. Hlavním přínosem tohoto standardu je specifikace jednotné struktury ovladače. V této struktuře jsou implementovány logické funkční bloky společné pro libovolný typ přístroje. Každý z bloků je vybaven příslušnou skupinou funkcí, která je přístupná z aplikačního programu [3].

Pro implementaci ovladače v systému Microsoft Windows je požadována DLL knihovna v kompilovaném tvaru (\*.lib) a hlavičkový soubor knihovny (\*.h). V hlavičkovém souboru lze najít obecnou deklaraci funkcí pro komunikaci s přístroji. Oba tyto soubory je nutné k aplikaci připojit.

Jednou z nejdůležitějších částí standardu je knihovna VISA (Virtual Instrument Software Architecture), která zajišťuje komunikaci mezi aplikačním programem a fyzickým zařízením. Tato knihovna umožňuje sestavovat programy takovým způsobem,



že jsou použitelné pro jakéhokoli výrobce. V knihovně `visa.h`, která je nainstalována společně s ovladači, lze najít všechny potřebné funkce pro ovládání přístrojů.

Zde uvádím funkce, které jsem při programování aplikace použil a stručně vysvětlím jejich vlastnosti. Jako parametry předávám funkcím mnou nadefinované proměnné. Tyto funkce lze najít v knihovně `visa.h`.

- `viOpenDefaultRM(&drm_session)`

Inicializování systému VISA a získání spojení (tzv. session).

- `viOpen(drm_session, ("GPIB0::" + IntToStr(Adr_Pri) + ">::INSTR").c_str(), VI_NULL, VI_NULL, &vi_session)`

Tato funkce slouží k navázání spojení se samotným přístrojem. K identifikaci VISA ovladače je použit parametr `drm_session`. Druhý parametr `("GPIB0::" + IntToStr(Adr_Pri) + ">::INSTR").c_str()` je textový řetězec s popisem volaného cíle (definuje, že chceme navázat komunikaci po GPIB s přístrojem na adrese XY). Další dva parametry zatím nejsou definovány (`VI_NULL`) a poslední parametr (`&vi_session`) je proměnná představující odkaz na spojení s přístrojem.

- `viClose(drm_session)`

Funkce pro ukončení komunikace.

- `viPrintf(vi_session, "*RST\n")`

Tato funkce provádí formátovaný zápis, vysílá přístroji data.

- `viQueryf(vi_session, "*IDN?\n", "%t", Q_buffer)`

Tato funkce provádí formátovaný zápis, vysílá přístroji data a zároveň požaduje odpověď. V tomto případě se jedná o vyžádanou identifikaci přístroje (`*IDN?`), odeslání textového řetězce přístrojem a zapsání obsahu řetězce do proměnné `Q_buffer`.

- `ViStatus close;`

`ViStatus` je proměnná, která může nabývat hodnoty `VI_SUCCESS`. Většina funkcí z knihovny VISA, při správném průběhu funkce vrací právě tuto proměnnou jako svou návratovou hodnotu. Toho se využívá při testování správného průběhu funkce.

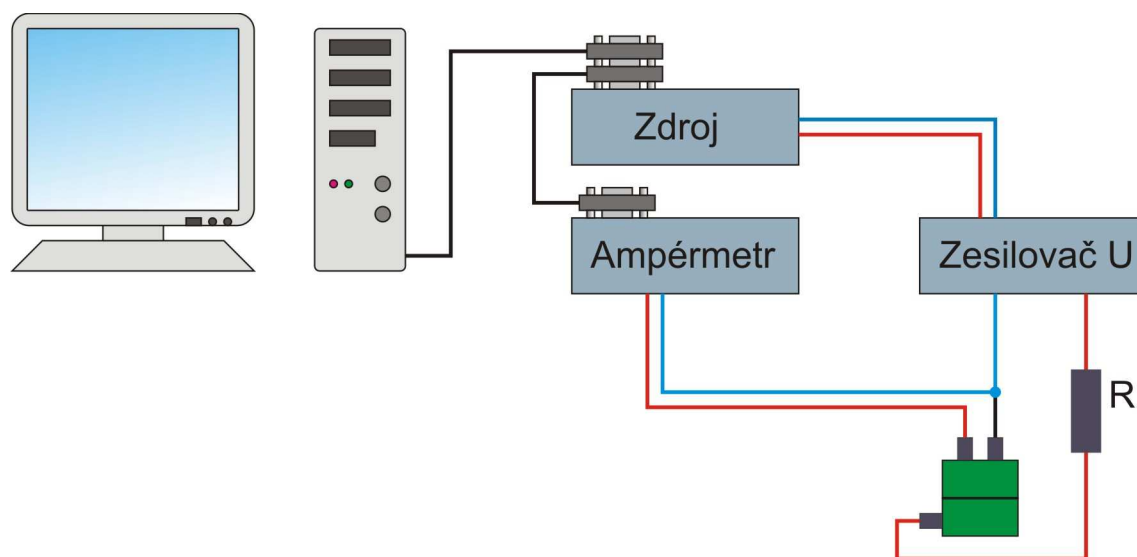
- `ViSession drm_session, vi_session;`

`ViSession` je typ, který definuje komunikační kanály v systému VISA.

### 3 Měřicí aparatura

Před vlastní realizací aplikace bylo nutné zvolit přístroje, které budou vhodné pro ovládání přes sběrnici GPIB a budou vyhovovat požadavkům měření. Tyto požadavky vzešly z ročníkového projektu [2] a jsou zmíněny v kapitole 1. Pro realizaci měření je tedy nutné zvolit zdroj napětí, voltmetr a ampérmetr.

Jako zdroj napětí jsem zvolil měřicí stanici AGILENT 34970A s modulem 34907A. Tento modul je vybaven analogovým výstupem. Jeho výstupní napětí je 12 V při maximálním proudu 10 mA. Tento zdroj napětí slouží zároveň jako voltmetr. Jako ampérmetr jsem použil multimetr KEITHLEY 6514, který dokáže měřit proudy v řádech pA, což je pro tento typ měření nezbytné. Oba tyto přístroje budou ovládány přes sběrnici GPIB (viz kapitola 2.1). Jak je popsáno v kapitole 1, pro měření izolačního odporu izolací je zapotřebí zdroj vysokého napětí (řádově jednotky kV). Pro dosažení tohoto napětí byl používán násobič napětí, který  $500 \times$  znásobil přivedené napětí z manuálně ovládaného zdroje. Pro ovládání tohoto násobiče napětí nám nyní slouží výše zmiňovaná měřicí ústředna AGILENT. Základní informace o původním zapojení měřicí aparatury lze najít v kapitole 1. Ilustrativní schéma zapojení pracoviště je uvedeno níže (Obrázek 3.1).



Obrázek 3.1 – Zapojení měřicí aparatury

Při ovládání zdroje napětí pomocí sběrnice GPIB je možné docílit přesnějšího nastavení napětí oproti manuálnímu nastavení. Díky tomu můžeme hodnotu napětí

nastavenou na zdroji brát jako hodnotu napětí na vzorku (samozřejmě po odečtení úbytku napětí na ochranném odporu).

Toto zjednodušení nám umožnilo vynechat voltmetr v měřícím obvodu. Obvod se tedy skládá ze zdroje napětí, který je připojen na zesilovač napětí. Jedna svorka zesilovače je připojena přes ochranný odpor k měřicímu přípravku (přípravek má zelenou barvu, viz Obrázek 3.1). Přes ampérmetr je obvod uzavřen na druhou svorku zesilovače napětí. Z měřícího přípravku je vyvedena ochranná elektroda, která je vyvedena mimo ampérmetr, aby byl měřen pouze proud, který prochází vzorkem (funkce ochranné elektrody je detailně popsána v literatuře [2]).

Fotografii realizovaného pracoviště lze nalézt v příloze A.

## 4 Realizace aplikace

Jako programovací software jsem zvolil C++ BUILDER od společnosti BORLAND. Důvodem, proč jsem si tento program zvolil, byla analogie s programem DEPLHI, se kterým jsem se seznámil v rámci dvou předmětů v nižších ročnících. To mi umožnilo rychlejší orientaci v programu. Dalším důvodem byly bohaté zkušenosti vedoucího diplomové práce a konzultanta s tímto programem a tím bylo možné předejít neočekávaným problémům s neznámým softwarem.

Pro komunikaci s přístroji po sběrnici GPIB bylo nutné nainstalovat ovladače pro převodník s rozhraním USB/GPIB. Součástí tohoto ovladače je i knihovna VISA, jejíž funkce je popsána v kapitole 2.2.2.

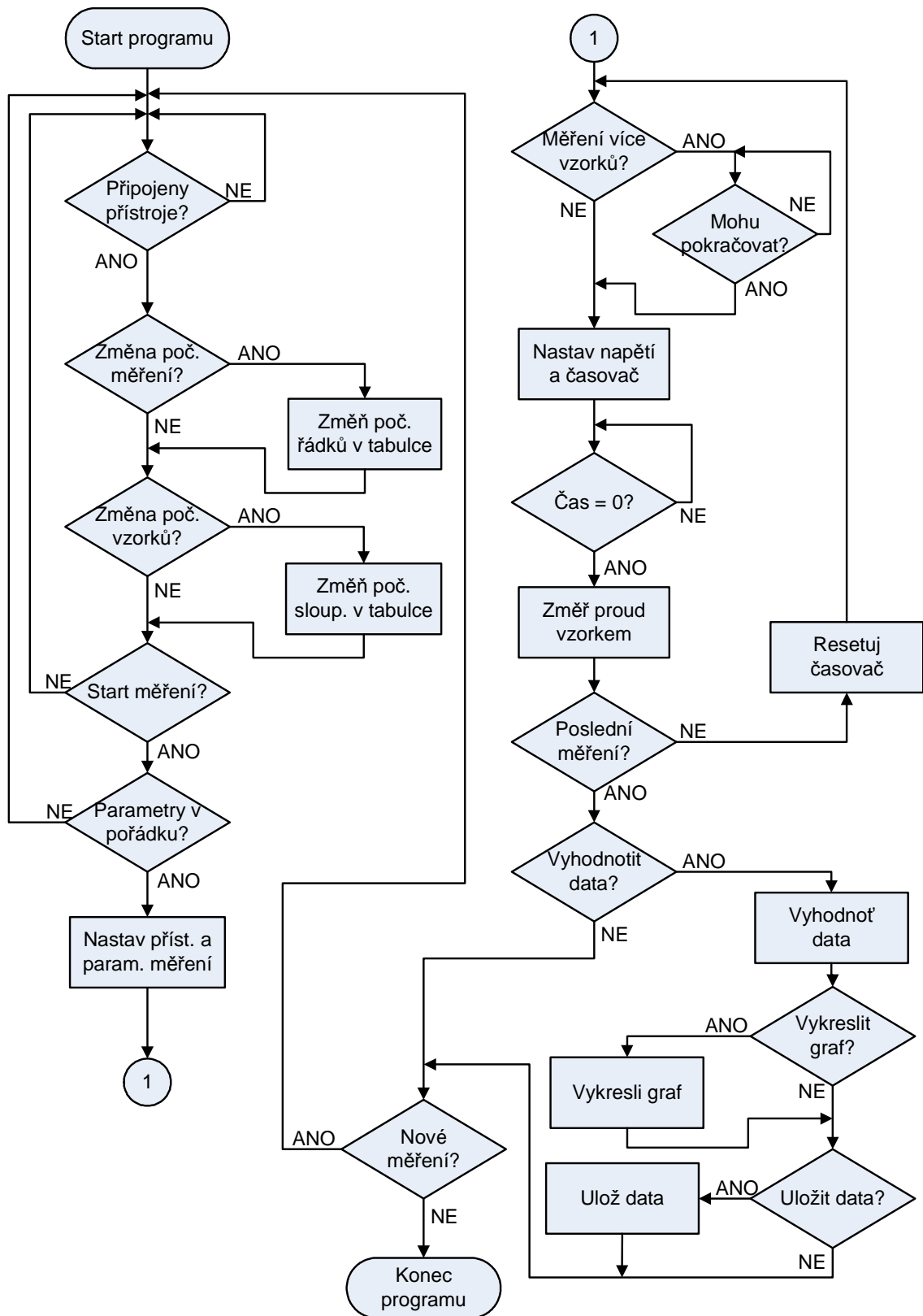
V následujících kapitolách uvedu návrh aplikace pomocí vývojového diagramu, definici tříd pro obsluhu jednotlivých zařízení, popis vzhledu a funkce celé aplikace.

### 4.1 Funkce aplikace

V této kapitole popíši základní funkce aplikace pomocí vývojového diagramu. Aplikace musí uživateli umožnit jednoduché nastavení parametrů měření a změření vzorku(ů) izolací dle zadaných kritérií. Z těchto předpokladů jsem si určil následující požadavky, které by aplikace měla umožňovat:

- Připojení přístrojů dle zadané adresy, jejich identifikaci a odpojení.
- Nastavení počtu vzorků k změření.
- Nastavení počtu a velikostí napětí, při kterých budou vzorky změřeny.
- Volbu času, který je potřebný pro ustálení proudu ve vzorku po přivedení napětí.
- Nastavení parametrů přístrojů .
- Automatické měření dle nastaveného napětí a času ustálení.
- Volbu pro záložní ukládání dat online při měření.
- Zobrazit tabulku s naměřenými hodnotami jednotlivých vzorků.
- Zobrazit tabulku hodnot po statistickém vyhodnocení měření s možností uložení.
- Grafické zobrazení výsledků s možností uložení.

Z těchto požadavků jsem nakreslil vývojový diagram (Obrázek 4.1), z něhož je patrná zjednodušená funkce programu.



Obrázek 4.1 – Vývojový diagram aplikace

Toto bylo nastínění funkce celé aplikace z hlediska uživatele, při nejjednodušší variantě měření.

Pro měření je nejprve nutné připojení k oběma přístrojům. Jako další krok následuje zadání základních parametrů měření. Při nejjednodušší variantě měření postačí zadat pouze hodnotu napětí. Samotné měření pak bude probíhat v daných intervalech, vždy po uplynutí doby nutné k ustálení proudu. Při měření více vzorků je vždy po proměření jednoho vzorku program přerušen a čeká na uživatele, který dá pokyn k dalšímu měření. K přerušení měření dochází z důvodu výměny vzorků. Po ukončení měření je možné provést statistické vyhodnocení naměřených hodnot a uložení výsledků. Po vyhodnocení lze výsledky zobrazit v grafu. Dále je možné začít nové měření stisknutím tlačítka Nové měření, přičemž všechny nastavené parametry zůstanou zachovány.

V následující kapitolách bude popsáno, jakým způsobem jsem aplikaci realizoval.

## 4.2 Třída Přístroj

V zadání práce je požadavek, aby byla možná jednoduchá úprava zdrojového kódu aplikace. Z těchto důvodů jsem vytvořil obecnou třídu `Přístroj`, ve které byly definovány základní metody a atributy pro obsluhu měřícího přístroje. Z této třídy byly odvozeny třídy `Ampermetr` a `Zdroj`. U těchto tříd byly dodefinovány specifické metody a atributy pro daný přístroj. Z těchto tříd pak byly vytvořeny jednotlivé instance.

Třída `Přístroj` je obecná, tzn. že obsluhuje jen základní funkce přístroje. Pomocí ní je možno se k přístroji připojit, odpojit, vyžádat si identifikaci a poslat přístroji data, resp. vyžádat si i odpověď. K tomu slouží i atributy definované v této třídě.

```
class Přístroj
{
private:
    char *Q_buffer;
    ViSession drm_session;
    ViSession vi_session;
public:
    int Address;
    bool IsConnected;
    __fastcall Přístroj();
```

```
__fastcall ~Pristroj();  
bool __fastcall open_prist(ViUInt32 Adr_Pri );  
bool __fastcall close_prist();  
AnsiString __fastcall ident();  
void __fastcall Posli_dataP(AnsiString DataP);  
AnsiString __fastcall Posli_dataQ(AnsiString  
DataQ);  
};
```

V části deklarace `private` je definován atribut `Q_buffer`, který využívají funkce vracející jako svou návratovou hodnotu typu `AnsiString`. Pomocí `ViSession` jsem nadefinoval `session` k identifikaci VISA (`drm_session`, `drm` – Default Ressource Manager) a `session` pro daný přístroj `vi_session`. K těmto položkám umístěným v části `private` lze přistupovat pouze uvnitř metod této třídy nebo v metodách tříd od této odvozené.

V části nazvané `public` jsou již definovány metody pro tuto třídu a atributy, ke kterým lze přistupovat odkudkoli. Definoval jsem zde proměnnou `Address`, kterou používám při zadání adresy přístroje. Tato proměnná se pak dále předává do funkce `open_prist`. Proměnná `IsConnected` slouží pouze k uchování informace o stavu připojení přístroje. Pokud je přístroj připojen, je `TRUE`, pokud ne, je `FALSE`. V dalším textu nastíním metody této třídy.

- `__fastcall Pristroj(), __fastcall ~Pristroj()`

Toto je deklarace konstruktoru (resp. destrukturu). Slouží pro inicializaci proměnných deklarovaných v části `private`. Při vytvoření instance od této třídy (`Pristroj`) se konstruktory automaticky zavolá, při jejím zániku se automaticky zavolá destruktory. V konstrukturu je inicializována proměnná `Q_buffer` takto: `Q_buffer = new char[100]`. V destrukturu je pak uvolněna paměť tímto způsobem: `delete[] Q_buffer`.

- `bool __fastcall open_prist(ViUInt32 Adr_Pri )`

Pomocí této metody naváží spojení s přístrojem a to tak, že v této metodě zavolám funkci `viOpen`. Jako parametr předávám požadovanou adresu přístroje, kterou si zvolíme v rozsahu 0-30. Funkce `viOpen` po úspěšném připojení vrací hodnotu `VI_SUCCESS`, kterou testuji. Na základě tohoto



porovnání vrací metoda `open_prist` `TRUE` nebo `FALSE`. Metoda je napsána tak, aby byla schopna se připojit pouze k zařízení na GPIB sběrnici. Jednoduchou změnou by bylo možné připojení i k jiným typům sběrnice. Při volání funkce `viOpen` by byl pouze pozměněn řetězec pro připojení k přístroji – viz. kapitola 2.2.2.

- `bool __fastcall close_prist()`

Tato metoda ukončí spojení s přístrojem a vrací hodnotu `TRUE`, pokud odpojení proběhlo v pořádku. V opačném případě vrací hodnotu `FALSE`. V této metodě je zavolána funkce `viClose`, která vlastní spojení ukončí. Funkce `viClose` vrací, stejně jako `viOpen`, hodnotu `VI_SUCCESS`, pokud proběhne v pořádku. Na základě této proměnné je nastavena návratová hodnota této metody.

- `AnsiString __fastcall ident()`

Tato metoda slouží k identifikaci přístroje. Její návratová hodnota je řetězec s informacemi o přístroji v podobě proměnné `Q_buffer`. V této metodě je volána funkce `viQueryf`.

- `void __fastcall Posli_dataP(AnsiString DataP)`

Tato metoda slouží k odesílání příkazů přístroji v podobě textových řetězců. Příkazy jsou ve formátu SCPI. Metoda zavolá funkci `viPrintf`, která řetězec pošle přístroji.

- `AnsiString __fastcall Posli_dataQ(AnsiString DataQ)`

Tato metoda slouží, stejně jako předchozí metoda, k odeslání příkazů přístroji, ale zároveň si vyžádá odpověď na tyto příkazy. Tímto způsobem se posílají příkazy typu „změř a pošli hodnotu“ nebo „identifikuj se“. Této metodě předávám jako parametr příkazy v podobě textového řetězce (SCPI). V metodě je zavolána funkce `viQueryf`, která příkazy pošle přístroji a vyžádá si odpověď. Tu uloží do proměnné `Q_buffer` a ta je pak předána jako návratová hodnota metody `Posli_dataQ`.

### 4.2.1 Třída Ampermetr

Třída `Ampermetr` byla odvozena od třídy `Pristroj`. To znamená, že zdělila všechny její metody a atributy. K nim byly přidány definice metod potřebných pro ovládání přístroje jako je ampérmetr. Zde uvádím deklaraci třídy `Ampermetr`, kde je detailně vidět, že je odvozena od třídy `Pristroj` a vidíme také její nové metody.

```
class Ampermetr : public Pristroj
{
private:

public:
    void __fastcall Inicializace();
    void __fastcall NastavRozsah (int Rozsah_index);
    void __fastcall NastavParam (int Index1,int Index2);
};
```

Jak můžeme vidět, definovány byly pouze tři nové metody. Ovšem tyto metody nám naprosto postačí pro ovládání ampérmetru podle našich požadavků.

- `void __fastcall Inicializace()`

Pomocí této metody přístroj, jak už název napovídá, inicializuji. To znamená, že přístroji pošlu příkaz, aby se nastavil do funkce ampérmetru (v našem případě se jedná o multimetr KEITHLEY 6415). Dále nastavím formát přijímaných dat na ASCII (je také možnost přijímat tyto data jako REAL, tj. mantisu s exponentem), tzn. že přijatá data budou ve formě textového řetězce. Další nastavení, které budu po přístroji požadovat, se týká dat vyslaných přístrojem zpět k PC.

Přístroj může posílat zpět naměřená data, čas změření (počítá se od nuly po zapnutí přístroje nebo po resetu) a stavovou informaci (jaké parametry byly při měření zapnuty a jaké funkce využity). V této aplikaci vyžadují pouze změřená data. Posledním nastavením, které využívám je nastavení počtu měření na 1 (pokud si vyžádám změření a poslání dat, tak se tento příkaz provede pouze jednou). Tato funkce je volána při začátku měření a všechny zde popsané parametry budou implicitně nastaveny a nelze je dále v aplikaci měnit.

- `void __fastcall NastavRozsah (int Rozsah_index)`

Tato metoda je volána také při začátku měření, ihned po inicializaci. Díky ní je možné nastavit rozsah ampérmetru na jednu z jedenácti možných hodnot včetně automatického rozsahu, který je nastaven přístrojem automaticky podle měřeného proudu. Při měření se ukázalo, že automatický rozsah není vhodný pro měření proudů v řádech desítek pA (byl používán rozsah 200 pA). Jako parametr této metody je předáván index rozsahu ampérmetru, který je vybrán z ComboBoxu. Podle tohoto rozsahu je vybrán příslušný řetězec (rozsah) a poslán pomocí funkce `Posli_dataP` ampérmetru. Nastavení rozsahu ampérmetru je v aplikaci umístěno v záložce Nastavení.

- `void __fastcall NastavParam (int Index1,int Index2)`

Tato metoda slouží k nastavení parametrů ampérmetru. Pro měření proudu je u ampérmetru možné nastavení ZCHECK (on/off) a ZCOR (on/off). Přesnější popis těchto parametrů ampérmetru lze najít v manuálu pro přístroj KEITHLEY 6514 nebo v nápovědě této aplikace. Metodě jsou předány dva parametry, které reprezentují indexy ve dvou komponentách RadioGroup a vyjadřují nastavení již zmiňovaných parametrů. Tyto parametry lze měnit v záložce Nastavení a jsou poslány přístroji ihned po inicializaci, při začátku měření.

#### 4.2.2 Třída Zdroj

Třída `Zdroj` byla stejně jako třída `Ampermetr` odvozena od třídy `Pristroj`. Zde bylo nutné vytvořit metodu, která umožňuje nastavení napětí, které je přivedeno na svorky násobiče napětí.

```
class Zdroj : public Pristroj
{
private:

public:
    void __fastcall NastavNapeti (float Hodnota);
};
```

- `void __fastcall NastavNapeti (float Hodnota)`

Tato metoda je volána pro nastavení napětí na zdroji. Jako parametr je jí předána hodnota napětí z tabulky, kam uživatel napíše hodnoty požadované pro měření. Z důvodu použití násobiče napětí (jak je zmíněno v kapitolách 1 a 3) bylo nutné zadávané napětí  $500 \times$  snížit, aby odpovídalo napětí nastavovanému na měřící stanici AGILENT.

### 4.3 Vzhled a funkce aplikace

Při tvorbě vlastní aplikace jsem se snažil o to, aby byl celý vzhled koncipován intuitivně, s důrazem na přehlednost celé aplikace. Chtěl jsem se vyvarovat toho, aby byl uživatel zmaten zbytečnou složitostí nastavení a nepřehledností aplikace. Z tohoto důvodu jsem program rozdělil do pěti záložek logicky jdoucích za sebou. V tomto pořadí je bude uživatel používat. Začne připojením přístrojů (Obrázek 4.2), následuje záložka Nastavení parametrů měření a přístrojů (Obrázek 4.3). V další záložce Měření zvolí hodnoty napětí, při kterých bude vzorek měřen a vlastní měření může začít (Obrázek 4.5). V této záložce jsou vidět hodnoty změřených proudů jednotlivých vzorků při daných napětích. Záložka Vyhodnocení (Obrázek 4.6) umožňuje statistické vyhodnocení měření, uložení výsledků a vykreslení grafu z těchto vyhodnocených hodnot. V poslední záložce je tento graf vykreslen (Obrázek 4.7). Zde může uživatel volit různá nastavení grafu jako grid a zoom nebo načíst již uložená data a zobrazit je.

Aplikaci jsem vytvořil tak, aby měla takzvané „zdvojené“ ovládání. Mimo tlačítek pro funkce, která jsou umístěna na jednotlivých záložkách, jsou tyto funkce duplikovány pomocí menu v horní části aplikace, jak je zvykem u většiny profesionálních aplikací. Naopak v dolní části aplikace je umístěn StatusBar. V něm se zobrazuje stav aplikace, zda jsou přístroje připojeny, zda probíhá měření apod. Pro přehled o čase je zde také uveden aktuální čas měření.

Celý program je koncipován tak, aby se v maximální míře zamezilo konfliktům mezi jednotlivými akcemi, které by uživatel mohl současně použít. Příkladem uvedu nemožnost měnit parametry měření a přístrojů při probíhající měření nebo nemožnost ukončení aplikace při probíhající měření. Aplikaci lze zavřít až po dokončení měření nebo po jeho zrušení.

### 4.3.1 Záložka Připojení

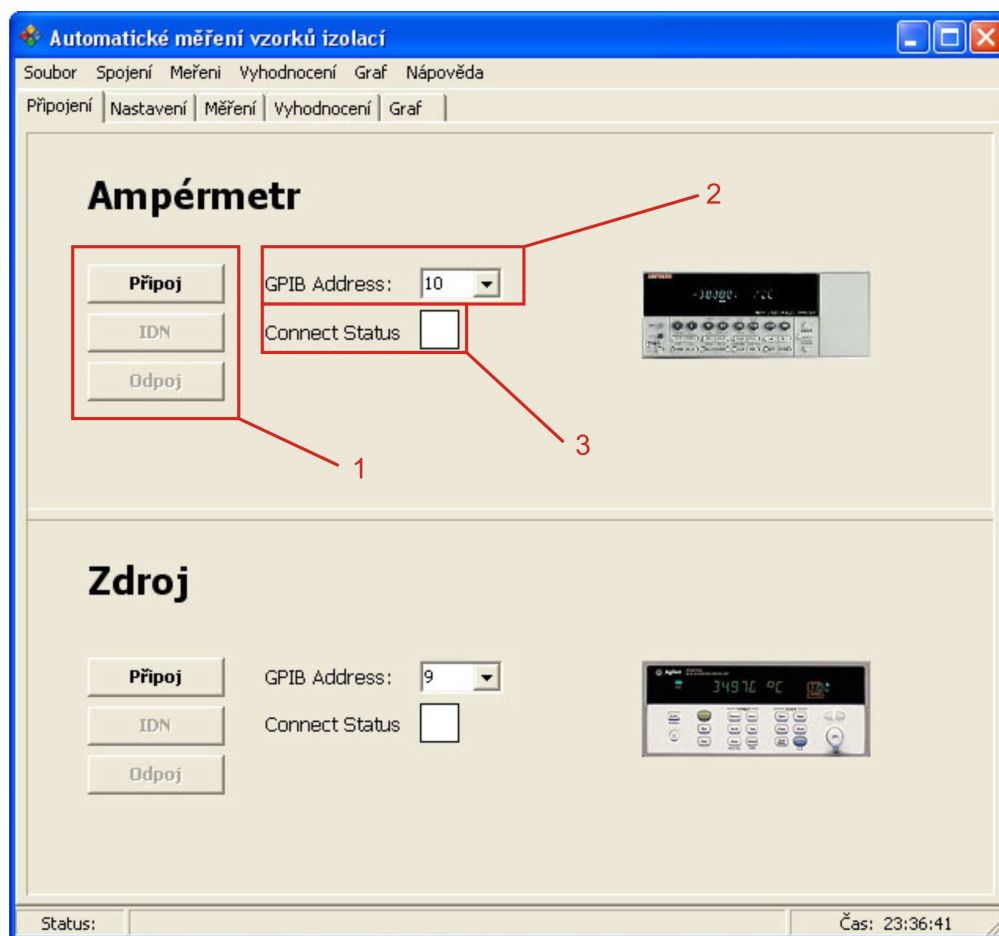
V této záložce je uživateli umožněno připojení, odpojení a identifikace obou přístrojů. Úspěšné připojení (po zmáčknutí tlačítka Připojit) je signalizováno zelenou barvou indikátoru (Connect Status), v opačném případě je indikátor červený a uživatel je vyzván ke kontrole spojení a k nastavení adresy přístroje. Při úspěšném odpojení bude mít indikátor bílou barvu. Pokud odpojení selže, indikátor bude červený. Adresu jednotlivých přístrojů může uživatel volit z nabídky v ComboBoxu (část č.2). Předvoleny jsou hodnoty, které jsem používal při ověřovacím měření. Uspořádání záložky Připojení můžeme vidět níže (Obrázek 4.2).

Zde je uveden popis jednotlivých částí:

Část č.1 – Tlačítka sloužící k připojení, odpojení a identifikaci přístroje.

Část č.2 – Volba adresy přístroje.

Část č.3 – Grafické znázornění připojení, resp. odpojení.

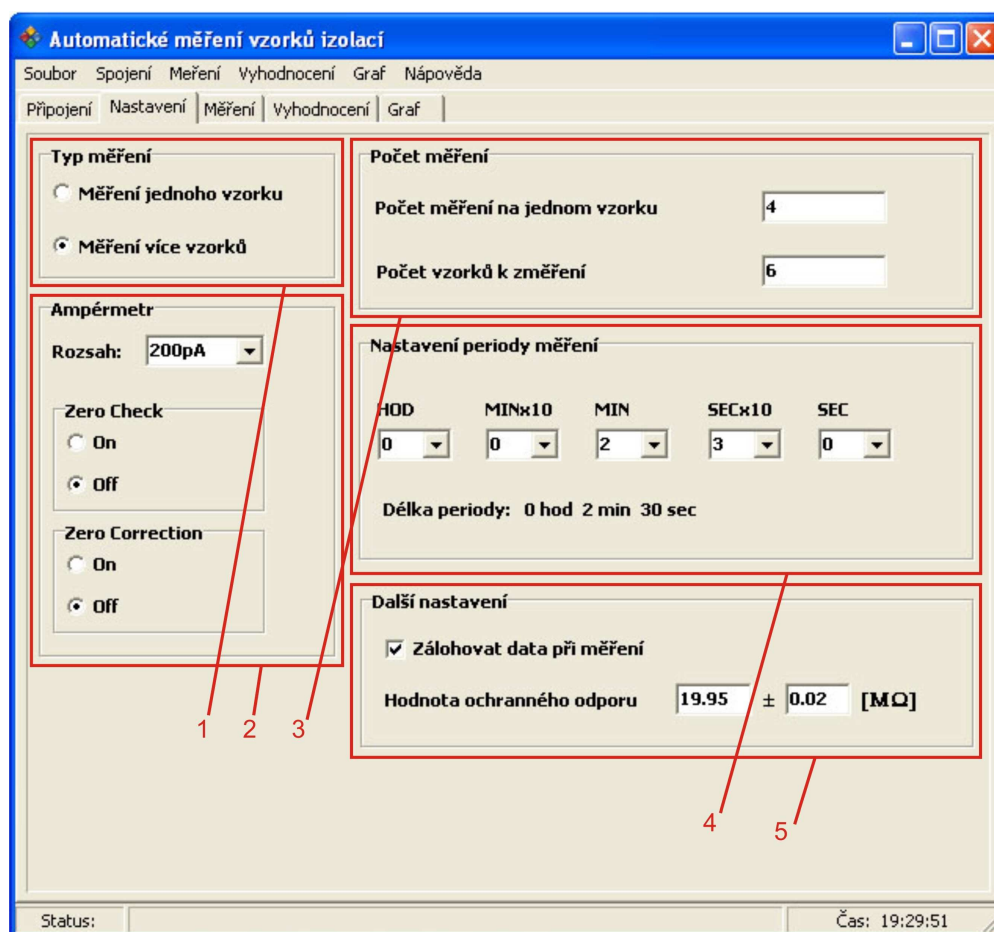


Obrázek 4.2 – Záložka Připojení

Pod pojmem identifikace přístroje si lze představit informace, které zadal do přístroje výrobce. Tyto údaje nám říkají o jaký typ přístroje se jedná, kdy byl vyroben apod. Po zmáčknutí tlačítka IDN budou tyto informace vypsány v novém dialogovém okně.

### 4.3.2 Záložka Nastavení

V této záložce uživatel nastaví parametry měření (zobrazení záložky viz Obrázek 4.3). Uživatel může ovlivnit nastavení ampérmetru, počet měření na jednom vzorku, počet vzorků celkem, čas pro ustálení proudu, uložení dat do záložního souboru, aj. Všechna tato nastavení jsou detailně popsána v následujícím odstavci.



Obrázek 4.3 – Záložka Nastavení

Detailní popis jednotlivých nastavení:

Část č.1 – Nastavení typu měření. Zde uživatel volí, zda bude předmětem měření pouze jeden vzorek, nebo jich bude více. Pokud zvolí více vzorků, aplikace mu automaticky umožní, po změření jednoho vzorku příslušným počtem

zvolených napětí, aby mohl měřené vzorky postupně vyměňovat. Pokud uživatel bude chtít změřit jeden vzorek vícekrát, pauza není potřeba a měření proběhne automaticky.

Část č.2 – Nastavení parametrů ampérmetru. Uživatel si může zvolit rozsah ampérmetru (standardně je nastaven automatický rozsah, který si ampérmetr nastaví podle procházejícího proudu). Zero Correction slouží k odečtení napěťového offsetu z měřené hodnoty a standardně je nastaven na OFF. Zero Check, pokud je zapnut (standardně je nastaven na OFF), zajistí změnu vstupního zesilovače pro odvedení vstupního signálu. Více podrobností k těmto nastavením lze najít v manuálu přístroje KEITHLEY 6514.

Část č.3 – Kolonka počet měření na jednom vzorku říká, kolika napětími bude jednotlivý vzorek změřen. Do kolonky počet vzorků k změření uživatel napíše kolik vzorků bude měřit. Pokud již dříve zvolil volbu Měření jednoho vzorku (volba je v části č. 1), bude zde kolonka Kolikrát změřit daný vzorek (viz. Obrázek 4.4 ).



Počet měření	
Počet měření na jednom vzorku	4
Kolikrát změřit jeden vzorek	6

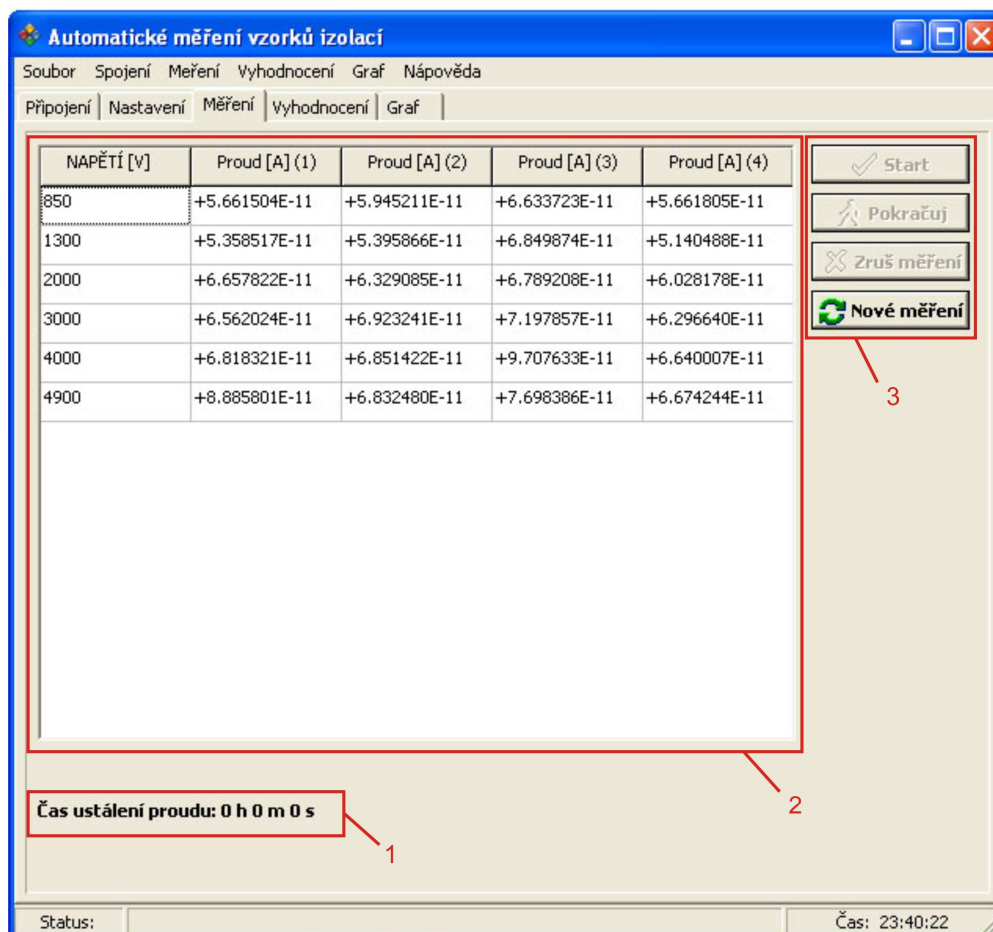
Obrázek 4.4 – Kolonka kolikrát změřit jeden vzorek

Část č.4 – Pomocí ComboBoxů si uživatel zvolí, jak dlouhý časový interval požaduje pro ustálení proudu procházejícího vzorkem. Je to doba od nastavení napětí na vzorku po změření proudu jím protékajícím. Je zde zobrazována také aktuálně zvolená hodnota intervalu.

Část č.5 – V této části může uživatel zvolit, zda chce při měření ukládat naměřená data do souboru. Tato data budou uložena pouze jako záložní (tzn. bez hlavičky kdy měření proběhlo a bez jakýchkoliv popisů hodnot), která uživatel nejspíše použije pouze v případě nečekaného přerušení měření (např. vypnutí proudu). Uživatel musí nastavit velikost ochranného odporu a jeho chyby v  $M\Omega$ . Jeho funkce je vysvětlena v kapitole 1.

### 4.3.3 Záložka Měření

V této záložce dochází již k samotnému měření. Uživatel do sloupce NAPĚTÍ vepíše požadované hodnoty napětí a měření může začít. Obrázek 4.5 zobrazuje uspořádání této záložky. Ve spodní části záložky vidí uživatel čas zbývajících do ustálení proudu procházejícího vzorkem (tzn. než bude proud změřen). Vpravo jsou tlačítka pro vlastní ovládání měření.



Obrázek 4.5 – Záložka Měření

Popis jednotlivých částí záložky Měření:

Část č.1 – Zde je zobrazen čas, který zbývá k ustálení proudu (resp. jeho změření).

Tato hodnota je každou sekundu inkrementována, tudíž uživatel má dostatečný přehled o délce probíhajícího měření.

Část č.2 – V této tabulce jsou zobrazovány změřené proudy při příslušné hodnotě napětí. Řádky a sloupce jsou automaticky generovány podle hodnot zadaných do položky Počet měření na vzorku, resp. Počet vzorků (pro



každý vzorek jeden sloupec pro hodnoty proudů). Do prvního sloupce s názvem NAPĚTÍ napíše uživatel požadované hodnoty napětí a zmáčkne tlačítko Start. Programově je zajištěno, aby uživatel nemohl zadat jiné znaky než čísla a výsledná hodnota nepřekročila maximální hodnotu napětí 5 kV.

Část č.3 – Tato skupina tlačítek pomáhá uživateli ovládat samotné měření. Pomocí tlačítka Start uživatel měření spustí. Další volbou, která je uživateli k dispozici, je tlačítko Zruš měření. Tímto tlačítkem je měření zrušeno a nelze v něm pokračovat (uživatel má možnost vyhodnotit i tato neúplná data).

Po zrušení měření nebo po jeho ukončení může uživatel použít tlačítko Nové měření. Zmáčknutím tohoto tlačítka jsou vymazána naměřená nebo již vyhodnocená data a aplikace je opět připravena k měření. To znamená, že všechny parametry měření, parametry přístrojů a hodnoty napětí zůstanou zachovány a uživatel může opět začít měřit zmáčknutím tlačítka Start.

Tlačítko Pokračovat je k dispozici pouze tehdy, pokud uživatel dříve zvolil nastavení Měření více vzorků (v záložce Nastavení). Následně vždy po změření jednoho vzorku (všemi napětími) je měření přerušeno. Napětí se nastaví na nulu a uživatel může bezpečně vyměnit vzorky v přípravku. Následné zmáčknutí tlačítka Pokračovat opět aktivuje měření.

Záměrně jsem mezi ovládací tlačítka měření nezařadil tlačítko Pauza. Je to z důvodu postupu měření. Pokud by uživatel přerušil měření, změnila by se doba ustálení proudu a tím by nebyla zajištěna opakovatelnost měření (resp. nebylo by možné porovnávat výsledky).

#### **4.3.4 Záložka Vyhodnocení**

V záložce Vyhodnocení jsou vyhodnoceny naměřené hodnoty. Tyto hodnoty lze uložit ve třech různých formátech. Uživatel aplikace může pro své potřeby připsat jednoduché poznámky k uloženým datům. Do souboru s daty se také uloží informace o velikosti ochranného odporu. Je zde také tlačítko Vykresli graf, které vyhodnocené výsledky převede do grafické podoby (což můžeme najít v kapitole 4.3.5). Obrázek 4.6

tuto záložku zobrazuje. Níže bude popsán rozdíl mezi jednotlivými formáty, do kterých lze vyhodnocená data ukládat.

- **\*.txt**

Tento formát je nastaven implicitně a jeho výsledkem je textový soubor s hlavičkou. V hlavičce je uloženo datum měření, čas měření, uživatelská poznámka k měření a hodnota ochranného odporu. Jednotlivá data jsou oddělena pomocí tabelátoru. Zde uvádím tabulku, která ukazuje, jakým způsobem jsou data v souboru uložena.

Zmereno dne: 6.3.2007 v 14:36:31 Popis vzorku: Zkusebni mereni..... Hodnota ochranneho odporu [MOhm]: 19.45 -----			
Napeti[V]	dU[V]	Proud[A]	dI[A]
1.5	0.03	1.4958E-7	0.0001E-7
2.3	0.03	2.2674E-7	0.0001E-7
3.33	0.03	3.2892E-7	0.0001E-7
5.8	0.04	5.7428E-7	0.0001E-7

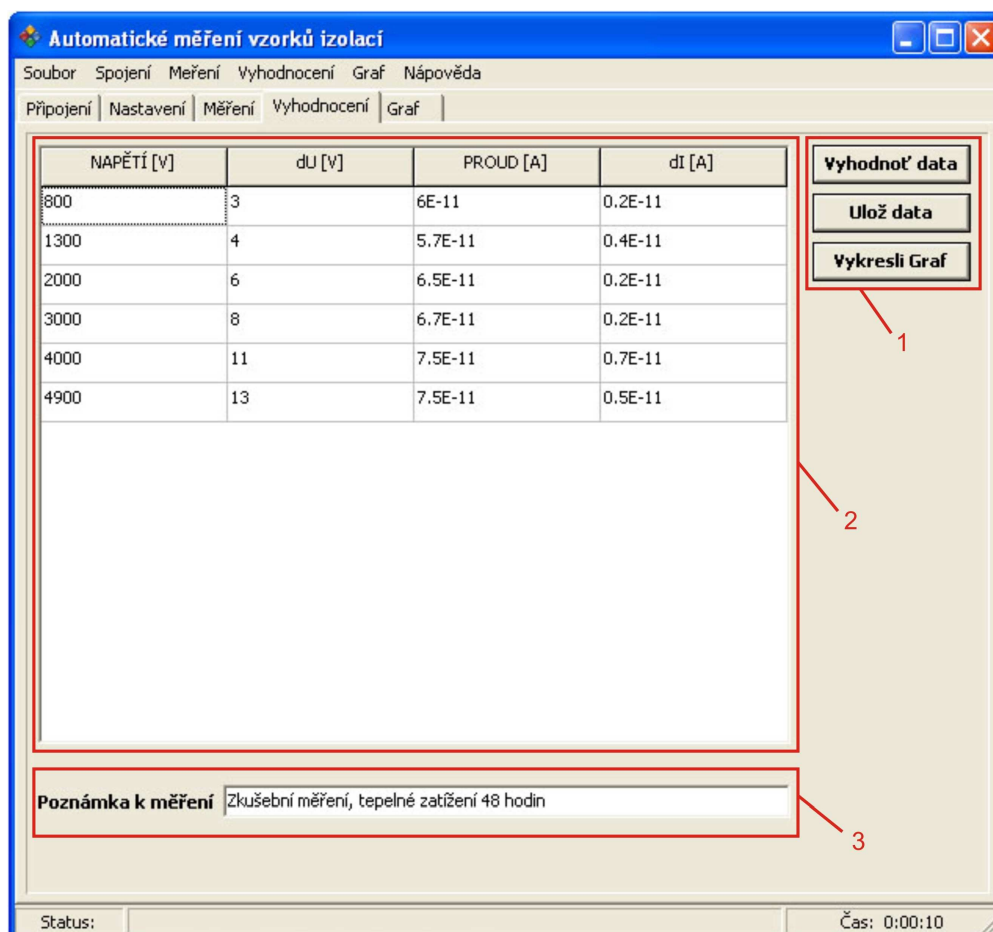
Tabulka 4.1 – Ukázka uložených dat v souboru \*.txt

- **\*.xls**

Tato přípona reprezentuje soubory programu Microsoft Excel. Díky uložení souboru s touto příponou může uživatel v programu Excel své výsledky rychle interpretovat a dále s nimi pracovat (vytvářet grafy, upravovat vzhled tabulek nebo grafů pro vložení do dalších dokumentů apod.).

- **\*.mdf**

Tento typ souboru jsem vytvořil pro jasné odlišení formátu dat uložených v tomto souboru. Jedná se o zkratku „Matlab Data File“. Z názvu vyplývá, že tato data budou vhodná pro načtení v programu MATLAB. Data jsou do souboru uložena bez hlavičky i bez popisu hodnot. Oddělovací znak mezi daty je mezera. Tento formát je pak vhodný pro načítání pomocí funkce `load` do již zmiňovaného programu MATLAB. Pokud bude uživatel chtít v aplikaci načíst data a graficky je zobrazit (viz kapitola 4.3.5), pak právě tento formát je jediný, který lze načíst.



Obrázek 4.6 – Záložka Vyhodnocení

Popis jednotlivých částí záložky:

Část č.1 – Tlačítko Vyhodnot' data slouží, po ukončení měření, k vyhodnocení naměřených hodnot zobrazovaných v tabulce na záložce Měření. Tlačítkem Ulož data uživatel vyhodnocená data uloží v jednom z formátů, které jsem v této kapitole popsal. Pokud uživatel měření přerušil tlačítkem Zruš měření, při ukládání dat se automaticky do souboru vloží informace, že měření nebylo dokončeno. K vykreslení grafu slouží tlačítko Vykresli graf. Graf je pak vykreslen na záložce Graf.

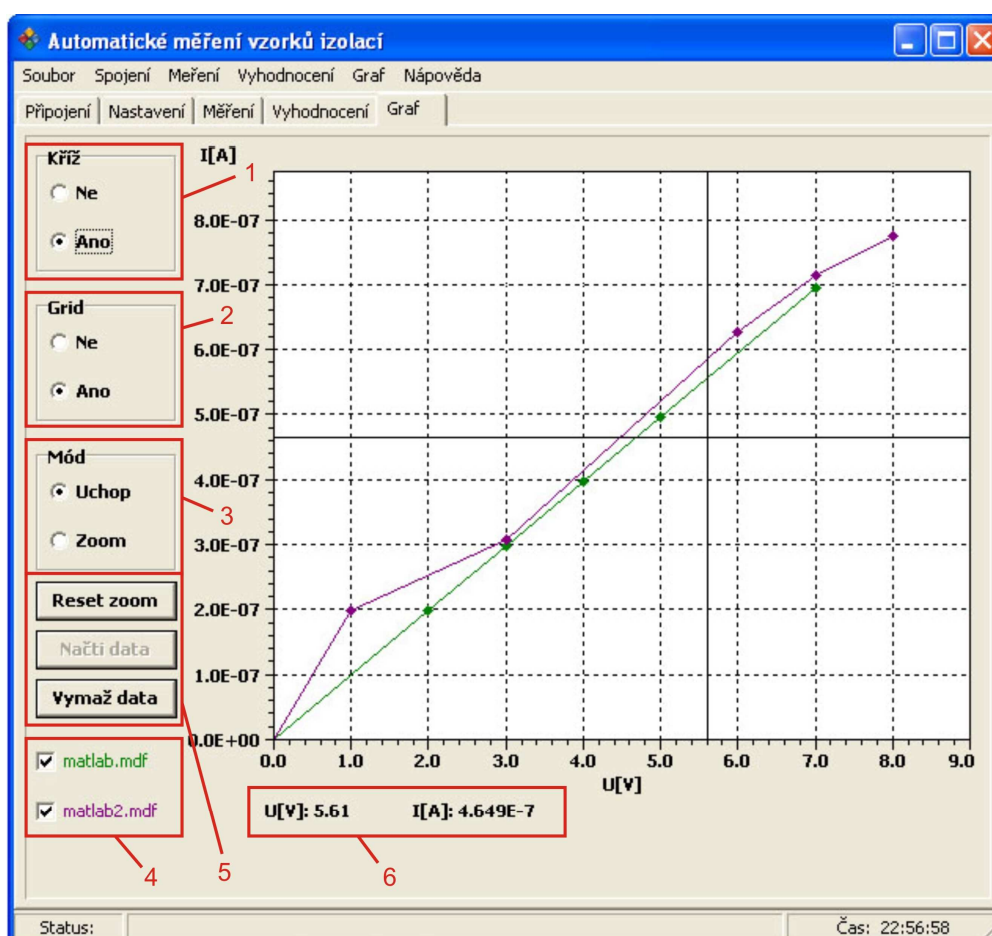
Část č.2 – V této tabulce jsou statisticky vyhodnocené výsledky napětí, proudů a jejich chyb. Každý řádek odpovídá jednomu napětí a proudu protékajícím jednotlivými vzorky. Řádky jsou generovány již při zadávání počtu měření na vzorku.

Část č.3 – Do této kolonky může uživatel napsat poznámku k měření, která může sloužit pro lepší orientaci v uložených datech. Tato poznámka je připojena k hlavičce ukládaného souboru.

Vyhodnocování dat je realizováno pomocí statistických metod. Více ke způsobu vyhodnocení lze nalézt v kapitole 6.1.

### 4.3.5 Záložka Graf

Záložka Graf slouží uživateli ke grafickému znázornění výsledků. Lze zde nastavit základní parametry grafu jako je grid, zoom nebo kříž, pomocí kterého lze lépe odečítat hodnoty z grafu. Uživatel také může načítat data ze souboru a následně porovnávat výsledky.



Obrázek 4.7 – Záložka Graf

Popis jednotlivých částí záložky :

Část č.1 – V této části uživatel zapne, popř. vypne kříž v grafu. Tento kříž slouží především k usnadnění odečítání hodnot proudu a napětí. Jejich hodnoty se obrazují v části č.6.

Část č.2 – Grid uživateli zpřehlední rozložení hodnot v grafu. Je možné ho, stejně jako kříž, zapnout nebo vypnout.

Část č.3 – Toto nastavení umožňuje zvolit si mód grafu. Uživatel má možnost graf přiblížit (Zoom) nebo posunout (Uchop). Tyto volby se vztahují na všechny průběhy zobrazené v grafu.

Část č.4 – V tomto místě jsou zobrazeny názvy souborů s daty, které uživatel načetl. Komponentou CheckBox je uživateli umožněno příslušné průběhy zobrazit nebo je skrýt. Barva textu názvu souboru odpovídá barvě průběhu vykresleného v grafu.

Část č.5 – Zde se nacházejí tlačítka umožňující práci s grafem. Tlačítko Reset zoom umožňuje po přiblížení grafu jeho návrat do plného zobrazení. Tlačítko Načti data slouží k načtení a zobrazení dat ze souboru. Lze načíst pouze soubory s příponou \*.mdf. Tento soubor lze vytvořit uložením vyhodnocených dat po měření. Po zmáčknutí tlačítka Vymaž data se vymažou všechny průběhy zobrazené v grafu.

## 4.4 Popis užitých komponent

V této kapitole popíšu komponenty, které jsem použil pro samotnou tvorbu aplikace. Popíšu zde také několik mnou vytvořených funkcí, které jsou v aplikaci použity. Všechny komponenty, vyjma jedné, jsou použity ze standardních knihoven dodávaných k vývojovému prostředí BUILDER C++. Pouze komponenta pro graf byla použita jako freeware získaný z internetu.

Vzhledem k rozsahu této práce se nebudu zabývat popisem všech komponent. Uvedu pouze ty nejdůležitější, které jsem použil v mé aplikaci a stručně popíši jejich funkci. Nejprve bych zmínil komponentu PageControl. Díky jejímu použití bylo možné aplikaci přehledně rozdělit do pěti záložek a pomocí „oušek“ se mezi jednotlivými záložkami přepínat.

Dále se zmíním o nevizuální komponentě ActionList. Pomocí této komponenty jsem vytvořil jednotlivé akce (jako je start měření, připojení apod.), které jsem přiřadil jednotlivým tlačítkům. Stejně akce jsem přiřadil i položkám v menu, pomocí kterého lze také celou aplikaci ovládat. Pokud chci, aby uživatel měl možnost např. spustit měření, tak tuto akci povolím. V opačném případě ji zakážu. Když je tato akce zakázána, znemožní se její spuštění jakýmkoli tlačítkem či položkou v menu. Díky této

komponentě bylo možné jednoduše vyřešit synchronizaci tlačítek v aplikaci a jejich odpovídajících náhrad v menu.

Menu bylo vytvořeno pomocí komponenty MainMenu a lze se v něm pohybovat myší i pomocí kláves. Pokud uživatel zmáčkne klávesu ALT, zobrazí se u jednotlivých položek v menu zkratky, pomocí kterých se může v menu pohybovat (např. pomocí kláves ALT+M se uživatel dostane do menu měření).

Pro zobrazení výsledných dat v grafu je použita komponenta RChart. V této verzi aplikace je pro graf použita komponenta s omezenou funkcí. Je to dáno tím, že je tato komponenta poskytována jako freeware. Z tohoto důvodu je možné zobrazení pouze dvou průběhů a nelze ukládat grafický výstup z grafu v podobě obrázku (např. bmp, jpg, ...). Výzkumný ústav mechatroniky a technické informatiky plánuje zakoupit pro další aplikace balíček komponent pro vykreslování grafů. Díky plné verzi bude graf již plně funkční.

Tuto komponentu lze nalézt na internetové adrese uvedené v literatuře [7], kde je možné ji stáhnout a nainstalovat do počítače (pouze jako freeware). Ta je pak zobrazena ve standardní nabídce komponent. Tuto komponentu jsem zvolil záměrně, protože komponenta Chart, která je standardně dodávána k programu BUILDER C++, byla nedostačující.

Komponenta RChart je přímo vyvíjena a určena pro zobrazování grafických výsledků a poskytuje celou řadu parametrů a nastavení pro práci s grafy. Díky této komponentě bylo možné neustále mapovat polohu kurzoru pro odečítání hodnot z grafu. Dále bylo možné graf po přiblížení posunovat. Tato nová komponenta také umožňuje, aby byl každý průběh samostatně upravován (zoom, posun atd.). Toto jsou jen některé příklady, v čem je mnou zvolená komponenta vhodnější. Tím, že je tato komponenta vyvíjena přímo pro grafické zobrazení dat, umožňuje velmi snadnou práci s naměřenými daty a poskytuje širokou paletu grafických úprav vykreslených průběhů.

Pro zobrazení naměřených dat v tabulkách je použita komponenta StringGrid.

## **4.5 Popis vybraných funkcí**

V této části práce bych rád popsal funkce, které jsem vytvořil pro vyhodnocování výsledků. Jedná se o funkce, které pomocí statistických metod

z naměřených hodnot spočítají výsledné hodnoty a provedou následnou úpravu čísel při jejich zobrazení v tabulce.

```
//Naplneni vysledne tabulky(pocitani hodnot a jejich chyb)
for (int j = 1; j <= PocRad ; j++) {
    if ((StrToFloat(edPocVzr->Text)==1)&&(cbRozsahAmper->ItemIndex!=0)){
        //nestatisticky
        Pole[j-1][2] = StrToFloat(stgTabMer->Cells[1][j]); //hod I
        Pole[j-1][3] = ChybaAmp(cbRozsahAmper->ItemIndex, Pole[j-1][2]);
    }else{
        //PROUD
        //prumerna hodnota proudu
        for (int i = 1; i <= PocSlp ; i++) {
            Suma = Suma + StrToFloat(stgTabMer->Cells[i][j]);
        }
        Prumer = Suma / PocSlp;
        Pole[j-1][2] = Prumer; //hodnota I do pole
        //smerodtna odchylka proudu
        for (int i = 1; i <= PocSlp ; i++) {
            Pom = StrToFloat(stgTabMer->Cells[i][j]) - Prumer;
            Suma2 = Suma2 + pow(Pom,2 );
        }
        Sigma = sqrt(Suma2/PocSlp);
        //chyba proudu zapsana do pole
        Pole[j-1][3] = Sigma/(sqrt(PocSlp));
    }
    //NAPETI
    //hodnota U zapsana do pole(zapocitan i ochranny odpor)
    Pole[j-1][0] = StrToFloat(stgTabMer->Cells[0][j]) - Pole[j-1][2]*
        StrToFloat(edHodnotaOdp->Text)*1E06;
    //chyba U zapsana do pole
    Pole[j-1][1] = (Pole[j-1][0]*0.0025 + 0.02)+
        ( StrToFloat(edHodnotaOdp->Text)*Pole[j-1][3]+
        Pole[j-1][2]*StrToFloat(edChybaOdp->Text) ); //R*dI+I*dR
}
```

Výše je uvedena funkce, pomocí které jsou statisticky vyhodnoceny naměřené hodnoty. Přesný popis statistických vzorců je možné najít v kapitole 6.1. Nejprve je z hodnot proudů při daném napětí spočten průměr (Suma). Tento průměr je brán i jako výsledná hodnota proudu. Dále je spočtena směrodatná odchylka (Sigma) a z těchto hodnot je spočtena výsledná chyba průměru. Výsledná velikost napětí je snížena o hodnotu úbytku napětí na ochranném odporu a poté je vypočítána chyba tohoto napětí jako chyba napětí nastavená na zdroji a chyba na ochranném odporu. Pokud je měřen pouze jeden vzorek, chyba změřeného proudu je spočtena dle vztahu (4) uvedeného v kapitole 6.1.

Dále se zaměřím na úpravu hodnot po jejich vyhodnocení. Ze zásad pro zapisování výsledných hodnot a jejich chyb vyplývá, že chybu zaokrouhlujeme nahoru na první platnou číslici a výsledek zaokrouhlujeme na stejný řád jako má tato chyba. I v této aplikaci bylo nutné se těchto zásad držet.

Bylo tedy nezbytné najít největší chybu proudu, zaokrouhlit ji na první platné místo. Podle této chyby provést korekci ostatních chyb a následně upravit i vlastní hodnoty proudů (resp. napětí) v tabulce výsledků. Po provedení prvních měření vyšlo najevo, že je dále nutné v tabulce výsledků upravit zobrazení hodnot proudů a jejich chyb. Aplikace zobrazovala například hodnotu proudu 1,235E-10 a jeho chybu 2E-13. Proto jsem musel upravit aplikaci tak, aby byla výsledná hodnota proudu a hodnota chyby zobrazena se stejným exponentem (např. hodnota 1,235E-10 a chyba 0,002E-10).

Z tohoto důvodu jsem naprogramoval následující funkci. Této funkci předávám pole, ve kterém jsou uložena vyhodnocená data, počet napětí (počet řádků), při kterých byly proudy měřeny a čísla sloupců (sloupec hodnoty a její chyby – A, B). V poli najdu největší chybu a podle ní určím parametr zaokrouhlování. Tento parametr určuje, jak se budou zaokrouhlovat ostatní čísla. Čísla následně zaokrouhlím (hodnoty podle klasických pravidel zaokrouhlování a chyby zaokrouhlím směrem nahoru). Poté se začnou upravovat hodnoty pro zobrazení. Zjistím tvar hodnoty proudu a podle něho upravím i jeho chybu do požadovaného tvaru. Celý tento proces probíhá tak, že zjistím velikost exponentu u hodnoty proudu a na tuto hodnotu upravím i exponent chyby. Dále je nutné vynásobit mantisu chyby příslušným násobkem tak, aby se nezměnila její hodnota a zároveň odpovídala tvaru příslušné hodnoty.

```
//reseni zaokrouhleni podle chyb
int ParZaokr;          //parametr pro zaokr (na kolik des mist)
double Max;            //pro maximalni hodnotu z pole
//najdu nejvetsi chybu
Max = Pole[0][A];
for (int i = 1; i < PocRadku; i++) {
    if (Max < Pole[i][A]) {
        Max = Pole[i][A];
    }
}

if (Max < 1) {          //kdyz najdu nejvetsi chybu <1, tak toto, jinak
ParZaokr=0
    double N = 10;
    ParZaokr = 1;
    //cyklus zjistí velikost cisla(poc nul) a nastaví par. Zaokrouhl.
    while (1) {
        if ( (Max*N)>1 ) {
            if (Max*N > 9) {      //reseni problemu cisla 0.09001
                ParZaokr--;
            }
            break;
        }
        ParZaokr++;
        N = N * 10;
    }
}
```



```

}
else ParZaokr = 0;
//vlastní zaokrouhlení čísel a jejich chyb podle pravidel
for (int i = 0; i < PocRadX; i++) {
    SetRoundMode (rmUp); //zaokrouhlím nahoru - chyby
    Pole[i][A] = RoundTo(Pole[i][A],-ParZaokr);
    SetRoundMode (rmNearest); //zaokrouhlím normalně - hodnoty
    Pole[i][B] = RoundTo(Pole[i][B],-ParZaokr);
    //srovnání čísel a chyb na stejný exponent (pouze pro proud)
    if (A==3) {
        int N = 1;
        AnsiString Pom,X,ExpHod,ExpChy;

        Pom = FloatToStr(Pole[i][B]); //pro exponent hodnoty proudu
        int pos = Pom.AnsiPos("E"); //pozice znaku v AnsiStr
        ExpHod = Pom.SubString(pos+2,1)+Pom.SubString(pos+3,1);
        //do ExpHod se napíše exponent

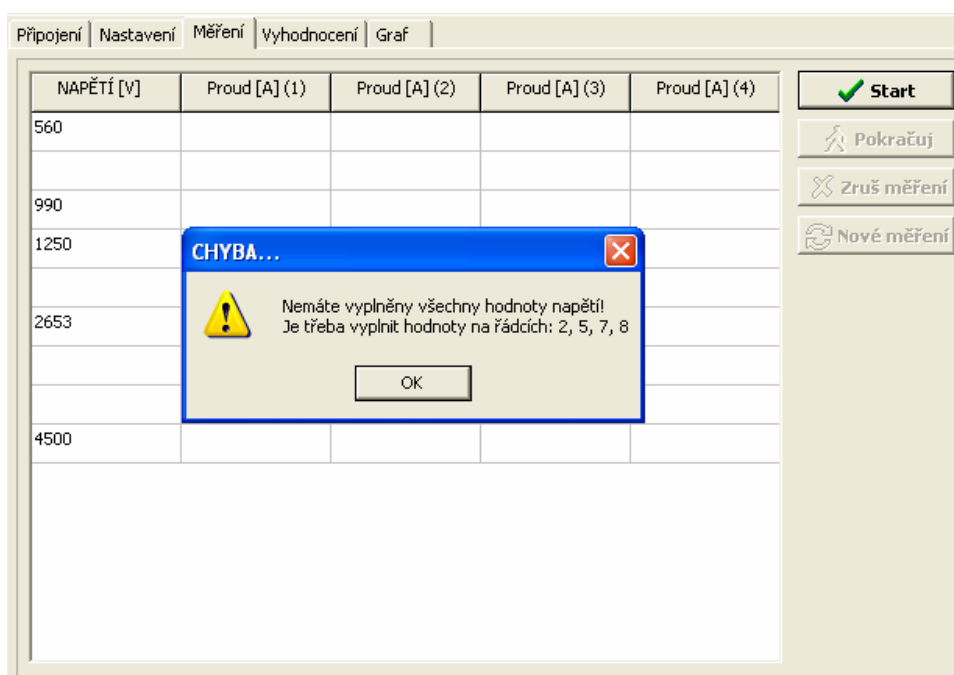
        Pom = FloatToStr(Pole[i][A]); //pro exponent chyby proudu
        pos = Pom.AnsiPos("E"); //pozice znaku v AnsiStr
        ExpChy = Pom.SubString(pos+2,1)+Pom.SubString(pos+3,1);
        //do ExpChy se napíše exp chyby proudu

        for (int i = 1; i <= Pom.Length()-4; i++) { //do X dám mantisu
            X = X+Pom.SubString(i,1);
        }
        for (int i = 1; i <= StrToInt(ExpChy)-StrToInt(ExpHod); i++) {
            N = N * 10; //spocítám jak mám upravit mantisu
        }

        stgTabTotal->Cells[A][i+1] = FormatFloat("0.#####",
            (StrToFloat(X)/N) ) + "E-" + ExpHod;
    }else stgTabTotal->Cells[A][i+1] = Pole[i][A];
    stgTabTotal->Cells[B][i+1] = Pole[i][B];
}

```

Celá aplikace je naprogramována tak, aby nedocházelo ke kolizím mezi jednotlivými akcemi (např. aby nebylo možné při měření ukončit aplikaci). Pokud by takováto kolize mohla nastat, uživatel je na tento problém upozorněn a aplikace neprovede požadovaný příkaz. V několika případech je uživatel vyzván k nápravě vzniklého problému. Jako příklad bych uvedl dialogové okno, které uživatele upozorní po zmáčknutí tlačítka Start na skutečnost, že nevyplnil všechna příslušná pole pro hodnoty napětí. Tento příklad vidíme na Obrázek 4.8.



Obrázek 4.8 – Ukázka ošetření uživatelských chyb

Tato upozornění jsou zobrazena například také při špatném připojení přístrojů nebo při zmáčknutí tlačítka Start bez připojených přístrojů.

## 5 Tvorba nápovědy

V praxi je obvyklé, že softwarové firmy dodávají ke každé komerční aplikaci nápovědu. Tato nápověda slouží uživateli k poznání všech možností aplikace nebo k upřesnění některých funkcí. Možnosti a funkce aplikace, které nejsou na první pohled jasné, má uživatel možnost najít v nápovědě i s přesným popisem jejich vlastností. Nápověda je také často opatřena příklady použití jednotlivých funkcí. V případě komplikovaného nastavení parametrů se jedná většinou o grafické znázornění (v podobě okna aplikace) a jeho popis.

Pro mou aplikaci jsem vytvořil nápovědu především proto, že zkoumání vlastností izolačního odporu izolací je velmi dlouhodobý proces, tudíž se předpokládá použití mé aplikace v dalších projektech dalšími uživateli. Tyto uživatelé nebudou znát všechny vlastnosti aplikace a tak především pro ně je nápověda určena. Zajisté i díky nápovědě obstojí má aplikace ve srovnání s profesionálními měřicími softwary.

Při tvorbě vzhledu aplikace jsem se řídil následujícími požadavky. Nápověda by měla umožňovat jednoduchou orientaci mezi tématy. Dále by neměla rušit svou výstředností nebo nevhodně volenými barvami, proto jsem se rozhodl pro světležlutý podklad nápovědy. Pro název tématu jsem vybral barvu šedivou a tzv. „ukotvení“. To znamená, že pokud bude dané téma delší než umožňuje zobrazit okno nápovědy, bude při scrollování vždy zobrazen název tématu.

### 5.1 Software pro tvorbu nápovědy

Po získání prvních informací o tvorbě nápovědy lze říci, že v současné době existují dva přístupy pro její tvorbu. Jedná se o tzv. WinHelp a HTML Help. Obě tyto možnosti se liší především vlastní tvorbou zdrojového textu a částečně i vzhledem výsledné nápovědy. Níže popíšu jejich některé základní vlastnosti.

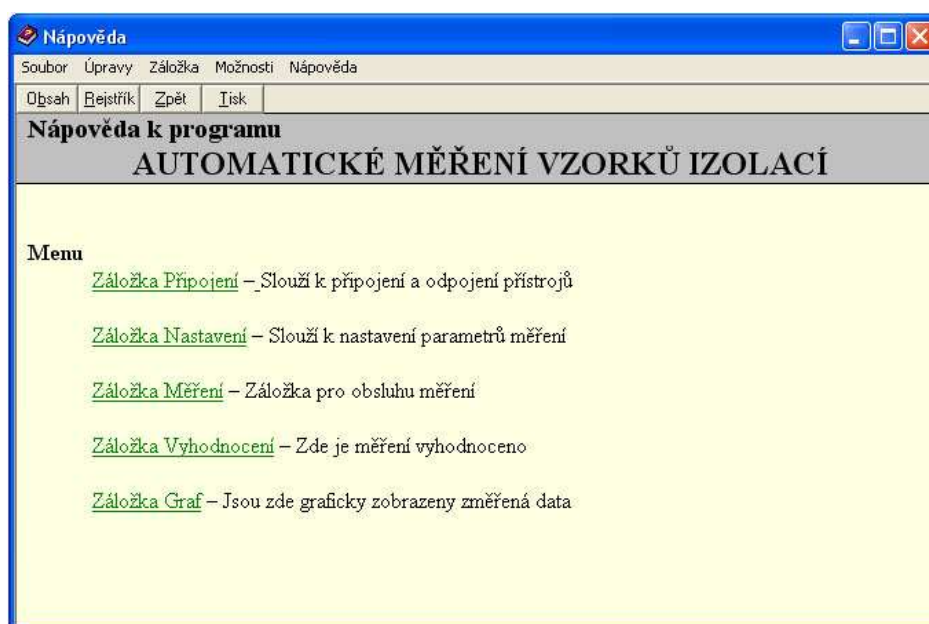
#### 5.1.1 WinHelp

WinHelp je systém pro vytváření nápovědy, který je starší než HTML Help, ale stále velmi hojně využíván. Hlavní výhodou pro tvůrce nápovědy, který nikdy nevytvářel např. HTML stránky, je její jednoduchá tvorba.

Jednotlivá témata je možné psát téměř v jakémkoli textovém editoru, který umožňuje ukládat soubor ve formátu \*.rtf a tvořit poznámky pod čarou. Drtivá většina uživatelů bude tento soubor vytvářet v prostředí Microsoft Word, kde je velice snadné měnit vlastnosti písma, odstavců, stylů atd.

Každé nové téma nápovědy se píše vždy na novou stránku a pro klíčová slova jsou použity tzv. footnote, což jsou poznámky pod čarou (např. #, \$, K), které mají pro kompilátor různé významy.

Pro tvorbu nápovědy je tedy nutné použít kompilátor, který zdrojový text vytvořený v \*.rtf formátu převede na soubor s příponou \*.hlp, což je již vlastní soubor nápovědy. Jako kompilátor slouží program Microsoft Help Workshop (hw.exe), který je volně ke stažení na internetu [8]. Obrázek 5.1 ukazuje základní vzhled tohoto typu nápovědy. Obsah nápovědy je umístěn v jiném okně než jsou zobrazena její jednotlivá témata. Pro jeho zobrazení stačí kliknout na tlačítko Obsah.



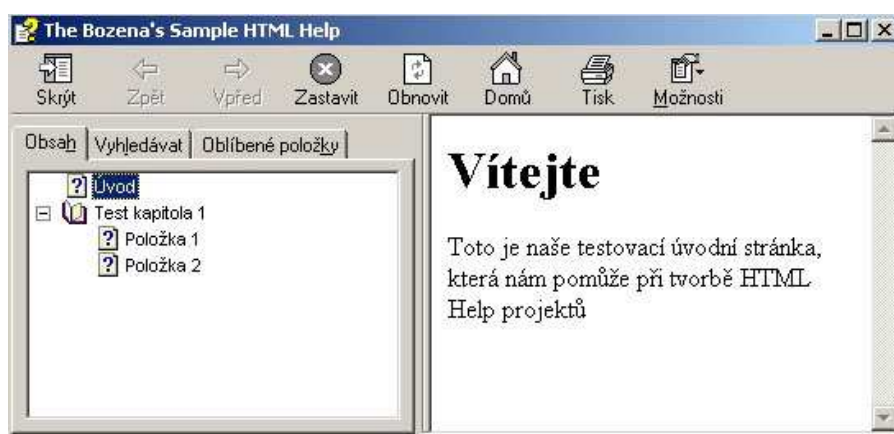
Obrázek 5.1 – Ukázka okna nápovědy WinHelpu

Tento druh nápovědy samozřejmě umožňuje mnoho dalších nastavení výsledného grafického zobrazení. Je například možné volit různé barvy pozadí či zabránit posouvání nadpisu tématu při scrollování. Možnostmi vzhledu a funkcemi nápovědy se v této práci více zabývat nebudu. Inspiraci pro tvorbu této nápovědy lze nalézt v literatuře [9].

### 5.1.2 HTML Help

HTML Help, jak již název napovídá, je založen na HTML kódu. Je to standardní nápověda, kterou lze použít pro Windows 98 s Internet Explorer 4.01 a vyšší verze těchto programů.

Jeho velkou výhodou je schopnost zobrazovat stránku nápovědy i obsah celé nápovědy v jednom okně. Díky tomu, že texty jsou psány v HTML kódu a ne v souborech typu \*.rtf (jako u WinHelpu), je výsledná velikost souboru s nápovědou menší. Je možné do něj implementovat i CSS styly a různé animace používané při tvorbě webů. Jako ukázka okna HTML Helpu je zde Obrázek 5.2. Pro tvorbu této nápovědy je nutné použít program htmlhelp.exe, který je šířen jako freeware (lze ho např. stáhnout z internetu [10]). V něm jsou jednotlivé stránky ve formátu HTML (i s obrázky) zkompileovány a je vytvořen výsledný soubor nápovědy s příponou \*.chm.



Obrázek 5.2 – Ukázka okna HTML helpu [11]

## 5.2 Realizace nápovědy

Pro samotnou realizaci nápovědy k aplikaci jsem zvolil nápovědu typu WinHelp. Její vlastnosti mé aplikaci naprosto postačí a její tvorba a připojení k aplikaci jsou velmi dobře popsány v literatuře [9]. Samozřejmě tedy bude připojení nápovědy k aplikaci. To znamená, že ji bude možné vyvolat pomocí menu aplikace. U jednotlivých funkcí, jako jsou tlačítka a různá nastavení, se dané téma nápovědy vyvolá klávesou F1.

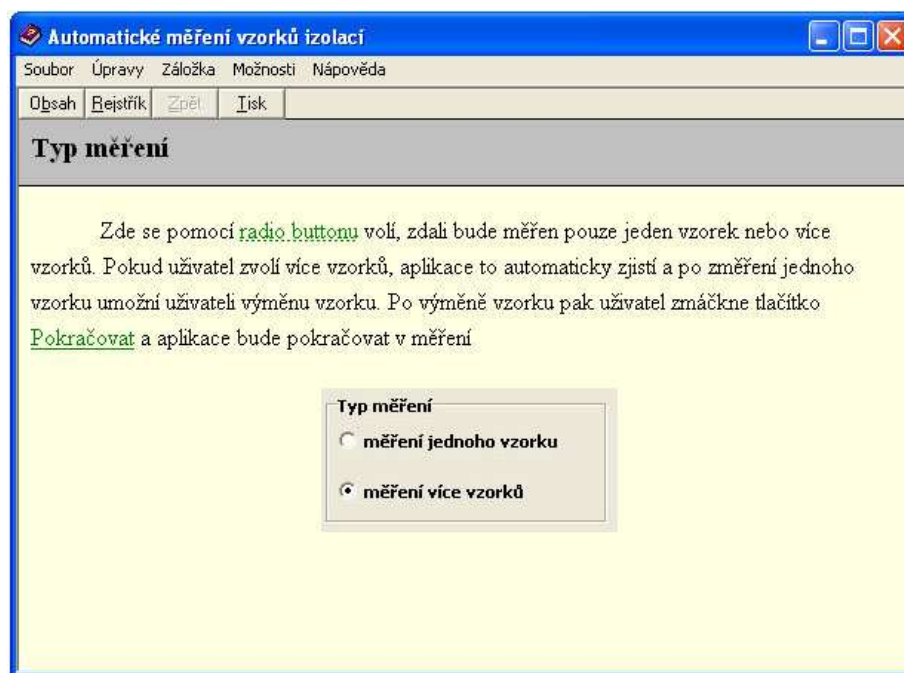
Nápověda je vytvořena přehledně a svou strukturou kopíruje strukturu aplikace. Hlavní menu nápovědy je rozděleno do pěti základních kapitol, stejně jako je pět záložek aplikace. Pod každou takovou kapitolou jsou umístěna témata k dané záložce.

Další pohyb v nápovědě je pak umožněn pomocí tzv. skoků, kdy uživatel klikne na zvýrazněné slovo a je přesměrován na téma odpovídající tomuto heslu. V nápovědě nechybí ani rejstřík, ve kterém jsou všechna klíčová slova uvedena a lze se mezi nimi jednoduše orientovat.

### 5.2.1 Témata nápovědy

Pro každou důležitou součást aplikace existuje v nápovědě samostatné téma. Celkem bylo vytvořeno 40 témat, které dostatečně postihují funkce a nastavení celé aplikace. Z každého tématu lze přejít do jiného (pokud je zde uveden odkaz – podtržený výraz) nebo se pouhým kliknutím na tlačítko Obsah vrátit zpět do obsahu. Pokud je odkaz podtržen tečkovanou čarou, po kliknutí na něj se téma neotevře na nové stránce, ale pouze v malém okně. V těchto případech se většinou nejedná o podstatné informace k funkcím aplikace, ale o informace typu „co je RadioButton“ a podobně.

Dále uvádím příklad nápovědy se zobrazeným tématem Typ měření. Na obrázku (Obrázek 5.3) je vidět, jak takové téma vypadá. Téma se skládá z názvu, který je umístěn v horní části okna a je v šedém pruhu a dále z vlastního popisu a řešení problému, který toto téma představuje. Některá témata jsou pro názornost doplněna obrázkem, který uživateli usnadní orientaci v problému.

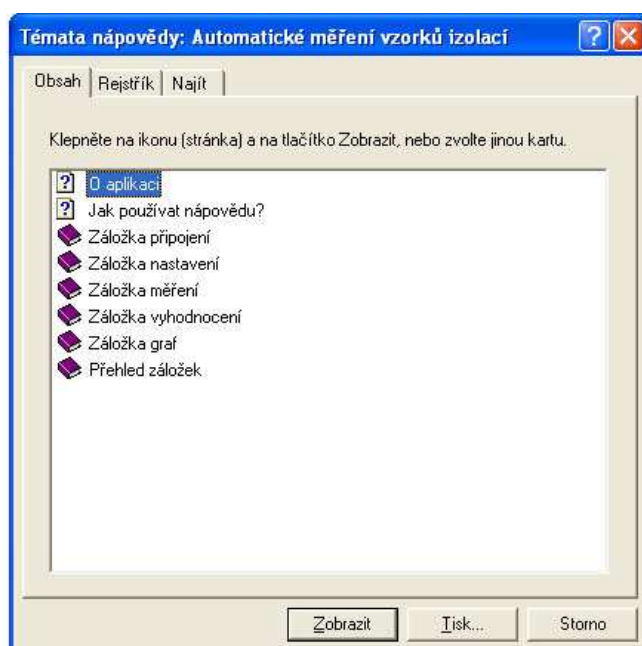


Obrázek 5.3 – Ukázka tématu nápovědy

### 5.2.2 Obsah a rejstřík nápovědy

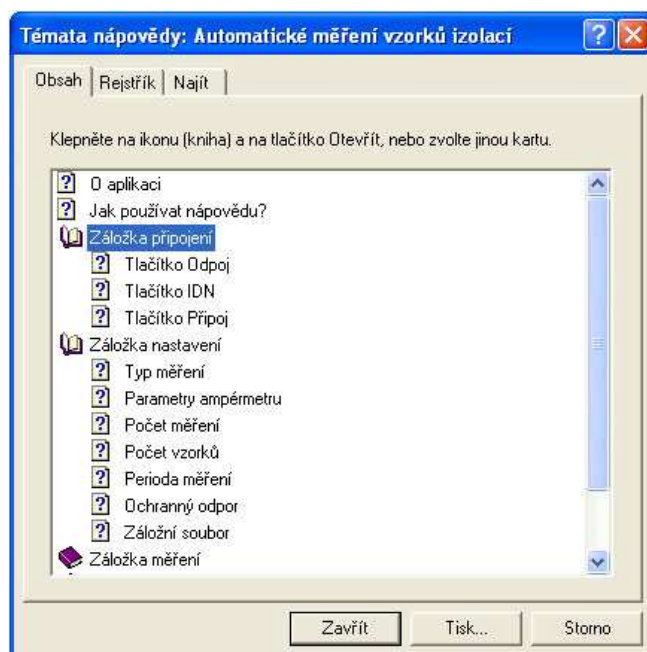
Po spuštění nápovědy je uživateli nabídnuto první téma (O aplikaci), které popisuje účel aplikace a prostředky potřebné k realizaci měření. Pro další orientaci v nápovědě uživatel vyvolá její obsah (Obrázek 5.4) zmáčknutím tlačítka Obsah.

Zde je první položkou právě zobrazené téma O aplikaci. Další položkou obsahu je téma Jak používat nápovědu?. V této části je vysvětleno, jak je nápověda koncipována a jakým způsobem lze nápovědu používat, aby uživatel dosáhl co nejrychlejší odpovědi na svůj problém.



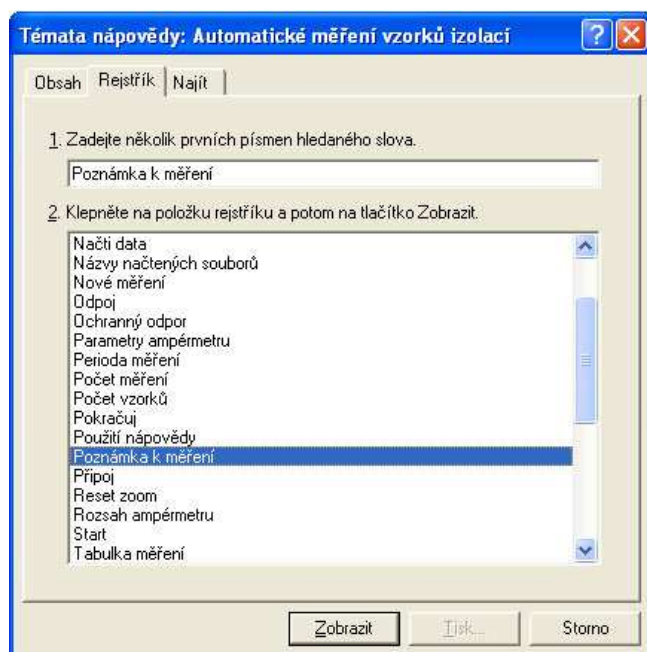
Obrázek 5.4 – Struktura obsahu nápovědy

Dále jsou v obsahu zařazeny jednotlivé kapitoly nápovědy. Tyto kapitoly, jak už bylo uvedeno výše, kopírují jednotlivé záložky aplikace. V každé této kapitole jsou umístěna další témata, vztahující se k dané záložce. Na následujícím obrázku (Obrázek 5.5) jsou vidět jednotlivá témata kapitol Záložka připojení a Záložka nastavení.



Obrázek 5.5 – Ukázka struktury témat v kapitolách

V případě, že uživatel nebude chtít využít obsahu děleného na kapitoly, může si klíčové slovo vyhledat pomocí rejstříku. V něm jsou obsažena všechna témata nápovědy. Podoba rejstříku je na obrázku níže (Obrázek 5.6).



Obrázek 5.6 – Rejstřík nápovědy



## 6 Ověřovací měření

K ověření správné funkce celé aplikace je nutné její použití při reálném měření. Teorie jak změřit izolační odpor izolantů již byla zpracována v literatuře [2], proto se jí zde nebudu podrobně zabývat. Přesné zapojení aparatury lze najít v kapitole 3. V této kapitole se budu soustředit pouze na způsob vyhodnocování výsledků. Dále zde budou prezentovány výsledky dosažené při měření.

### 6.1 Vyhodnocení výsledků

Ze zkušeností vyplývá, že při měření izolačního odporu se proud protékající vzorkem pohybuje v řádech pA a výsledná hodnota je zatížena velkou chybou. Proto je měření izolačního odporu vzorků izolací prováděno způsobem, že je vždy změřeno několik vzorků od stejného výrobce při stejném tepelném zatížení. Tyto výsledky jsou pak statisticky vyhodnoceny a jsou brány jako relevantní hodnoty pro daný vzorek při dané teplotě. Pro každý vzorek jsou samozřejmě nastaveny stejné hodnoty napětí, aby bylo možné výsledky měření porovnávat.

Připojení   Nastavení   Měření   Vyhodnocení   Graf					
NAPĚTÍ [V]	Proud [A] (1)	Proud [A] (2)	Proud [A] (3)	Proud [A] (4)	
850	+5.661504E-11	+5.945211E-11	+6.633723E-11	+5.661805E-11	Start
1300	+5.358517E-11	+5.395866E-11	+6.849874E-11	+5.140488E-11	Pokračuj
2000	+6.657822E-11	+6.329085E-11	+6.789208E-11	+6.028178E-11	Zruš měření
3000	+6.562024E-11	+6.923241E-11	+7.197857E-11	+6.296640E-11	Nové měření
4000	+6.818321E-11	+6.851422E-11	+9.707633E-11	+6.640007E-11	
4900	+8.885801E-11	+6.832480E-11	+7.698386E-11	+6.674244E-11	

Obrázek 6.1 – Tabulka naměřených hodnot

Jak můžeme vidět v tabulce naměřených hodnot (Obrázek 6.1), jsou na všech vzorcích změřeny proudy (sloupce označeny jako Proud [A] (1)...až Proud [A] (n)) při stejném napětí  $U[V]$ . Tyto hodnoty jsou statisticky vyhodnoceny a výsledky jsou uvedeny v tabulce vyhodnocených hodnot (viz kapitola 4.3.4). Hodnoty změřených proudů jsou vypočteny pomocí následujících vzorců.

Průměrná hodnota (výsledná hodnota proudu při daném napětí)

$$\bar{I} = \sum_{i=1}^N \frac{I_i}{N} \quad (1)$$

Směrodatná odchylka

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (I_i - \bar{I})^2}{N}} \quad (2)$$

Chyba průměrné hodnoty proudu

$$\Delta \bar{I} = \frac{\sigma}{\sqrt{N}} \quad (3)$$

Celé měření probíhá na ampérmetru při rozsahu 200 pA, proto jsou všechny hodnoty zatíženy přibližně stejnou chybou přístroje. Dále zde uvádím vzorec pro výpočet chyby ampérmetru pro rozsah 200pA.

$$\Delta I = \pm 1\% \text{rdg} \pm 5 \text{dgs} \quad (4)$$

Výsledná chyba ampérmetru se spočítá jako součet jednoho procenta ze změřené hodnoty a pěti jednotek na posledním desetinném místě daného rozsahu. Tento vzorec je použit pro vyhodnocení měření pouze tehdy, bude-li uživatel měřit vzorek pouze jednou, protože v tomto případě statistické vyhodnocení výsledků pozbývá smysl.

Výslednou hodnotu napětí na měřeném vzorku je nutné spočítat společně s ochranným odporem  $R$  jako dělič napětí. Výsledné napětí tak bude sníženo o napětí, které vyvolá proud protékající přes odpor. Při měření více vzorků je za hodnotu proudu ve vztahu (5) dosazena spočtená průměrná hodnota proudu. V případě měření jednoho vzorku je dosazen proud procházející vzorkem.

$$U_{VYSL} = U_{ZDR} - RI \quad (5)$$

Chybu napětí, která bude na měřeném vzorku, vypočítáme jako součet chyby zdroje a chyby napětí na ochranném odporu. Chybu zdroje spočítáme podle vzorce, který je uveden v manuálu měřicí stanice AGILENT 37970A s měřicí kartou 34907A a

chybu na ochranném odporu spočítáme dle zásad pro výpočty chyb měření. Jedná se o určení úplného diferenciálu uvažované funkce.

$$\Delta(I \cdot R) = \frac{\partial(I \cdot U)}{\partial I} \cdot \Delta U + \frac{\partial(I \cdot U)}{\partial U} \cdot \Delta I = I \cdot \Delta U + U \cdot \Delta I \quad (6)$$

Výsledkem je vztah č. 7 pro výpočet chyby napětí.

$$\Delta U_{VYSL} = (0,25\% \cdot U_{ZDR} + 20mV) + (R \cdot \Delta I + I \cdot \Delta R) \quad (7)$$

Všechny takto vypočítané hodnoty jsou uvedeny v tabulce s výsledky (viz kapitola 4.3.4). Tato data jsou brána jako výsledná data měření.

## 6.2 Výsledky ověřovacího měření

Pro ověření správné funkce aplikace jsem zvolil stejné měření vzorků jako v literatuře [2], které již byly proměřeny manuálním měřením. Jednalo se o 45 vzorků jednoho výrobce. Při samotném měření byly vždy změřeny čtyři vzorky při jednom tepelném zatížení a tyto hodnoty byly vyhodnoceny. Vzorky byly tepelně zatěžovány teplotou 175°C po dobu 8 až 80 hodin. Velikost ochranného odporu byla (19,95±0,02)MΩ. Přesnější informace o tepelném zatěžování jednotlivých vzorků lze nalézt v literatuře [2]. Dále zde uvedu, jak lze výsledky interpretovat například v programu MATLAB a pro přehlednost uvedu srovnání aktuálních výsledků s výsledky při manuálním měření.

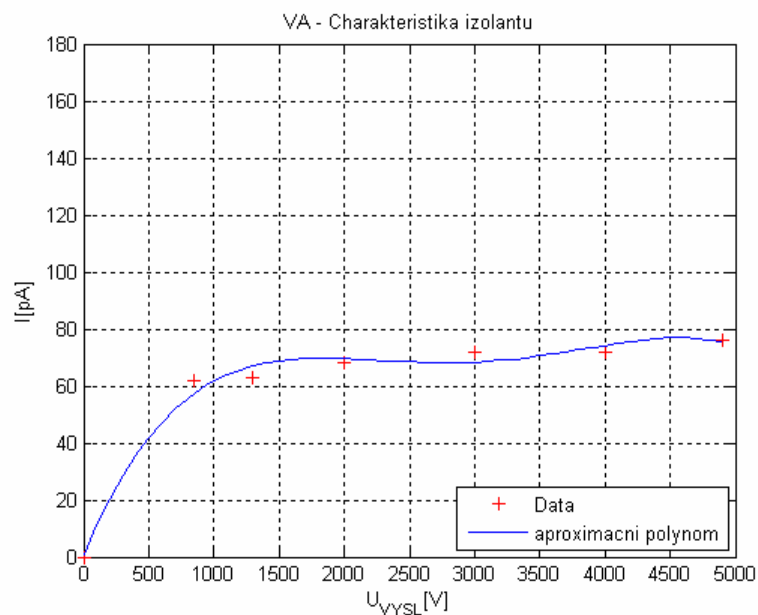
U[V]	I[pA]
850 ± 3	62 ± 2
1300 ± 4	63 ± 2
2000 ± 6	68 ± 2
3000 ± 8	72 ± 3
4000 ± 11	72 ± 1
4900 ± 13	76 ± 2

Tabulka 6.1 – Výsledné hodnoty vzorků tepelně zatížených po dobu 72 hodin

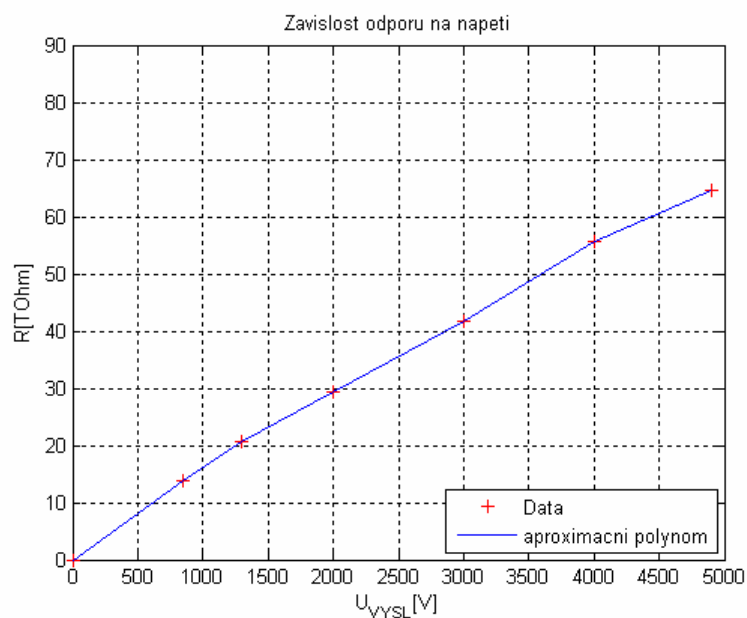
Rád bych zde uvedl upřesnění k hodnotám napětí, které jsem uvedl v tabulce (Tabulka 6.1). Napětí nastavené zdrojem musí být podle vzorce (5) sníženo o úbytek napětí na ochranném odporu, abychom dostali napětí na vzorku. Úbytek napětí na ochranném odporu je velmi malý, protože měřený proud je v řádu pA. Při spočtení

chyby a zaokrouhlení konečného výsledku se v hodnotě napětí tento úbytek neprojeví (ačkoli je s ním ve výpočtech počítáno).

Z této tabulky můžeme sestavit VA charakteristiku pro daná tepelná zatížení nebo graf závislosti odporu na napětí. Grafy jsou zpracovány v programu MATLAB a jsou zobrazeny níže.



Graf 6.1 – VA charakteristika vzorků tepelně zatížených po dobu 72 h



Graf 6.2 – Závislost odporu na napětí pro vzorky tepelně zatížené po dobu 72 h

Dále bych uvedl příklad porovnání výsledků manuálního a automatického měření. Jedná se o stejné vzorky se stejným tepelným zatížením o délce 72 hodin. Z tabulek můžeme vidět určitý pokles proudu při automatickém měření. Tento rozdíl je dán nehomogenními podmínkami měření (např. jiná vlhkost vzorku).

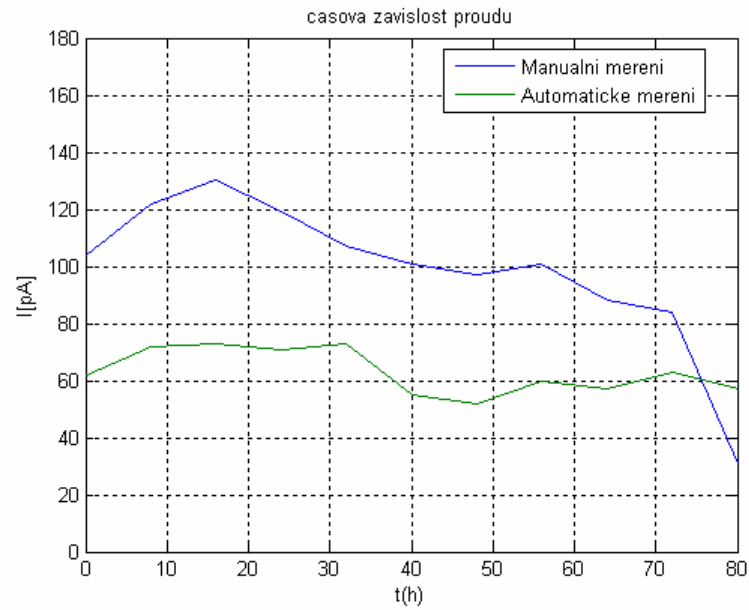
U[V]	I[pA]
878 ± 5	74 ± 2
1277 ± 8	84 ± 2
2033 ± 4	98 ± 2
3025 ± 7	104 ± 3
4024 ± 3	105 ± 4
4873 ± 4	106 ± 2

Tabulka 6.2 – Výsledky manuálního měření

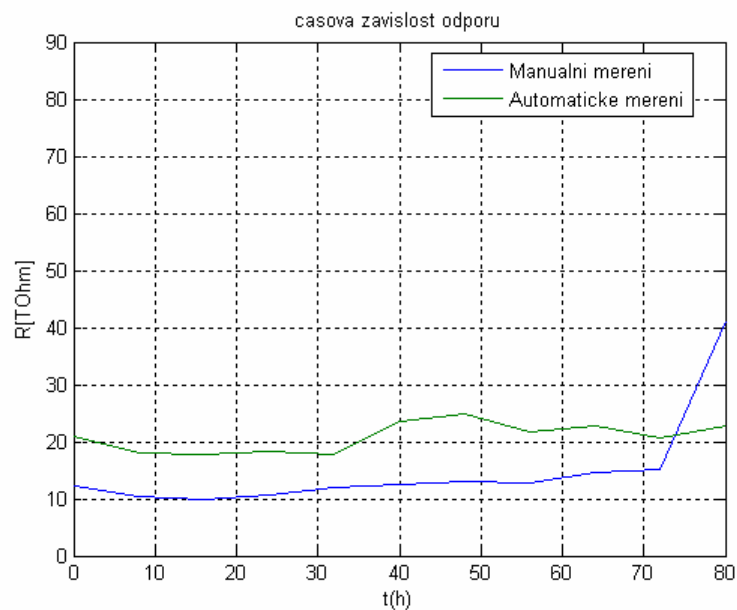
U[V]	I[pA]
850 ± 3	62 ± 2
1300 ± 4	63 ± 2
2000 ± 6	68 ± 2
3000 ± 8	72 ± 3
4000 ± 11	72 ± 1
4900 ± 13	76 ± 2

Tabulka 6.3 – Výsledky automatického měření

Pro další srovnání výsledků uvedu grafy, ve kterých je vidět vývoj proudu (resp. odporu) při rostoucím tepelném zatížení. Jako parametr je zvoleno napětí, při kterém byl daný proud měřen.



Graf 6.3 – Závislost proudu na délce tepelného zatížení (při napětí 1,3 kV)



Graf 6.4 – Závislost odporu na délce tepelného zatížení (při napětí 1,3 kV)

Na uvedených grafech jsou názorně vidět výsledky dosažené manuálním a automatickým měřením. Porovnání výsledků automatického a manuálního měření je obtížné z důvodu různých podmínek při měření.

## Závěr

Úkolem této diplomové práce bylo realizování aplikace pro automatické měření izolačního odporu izolačních materiálů, které se používají při výrobě asynchronních motorů. Bylo nutné vybrat vhodné rozhraní pro komunikaci PC a přístrojů, včetně přístrojů samotných. Součástí aplikace měla být i nápověda pro další uživatele. Vlastní teorie o měření izolačního odporu a zapojení přístrojů vycházela z ročníkového projektu [1].

Prostudoval jsem dostupná rozhraní pro komunikaci PC a měřících přístrojů. Pro svou variabilitu a snadné připojení k PC jsem zvolil rozhraní GPIB. Ve vývojovém prostředí BUILDER C++ jsem naprogramoval aplikaci pro ovládání měřících přístrojů a nastavování parametrů měření. Navržená aplikace umožňuje naměřená data vyhodnocovat, ukládat a graficky je zobrazit. Aplikace je naprogramována tak, aby s ní mohli jednoduše pracovat i nezkušení uživatelé. Všechny základní funkce aplikace jsou naprogramovány takovým způsobem, že je lze využít pro ovládání jakéhokoli přístroje. Funkce pro připojení, odpojení, posílání a přijímání dat lze využít pro libovolné přístroje, které komunikují přes sběrnici GPIB.

Vytvořil jsem nápovědu, která obsahuje veškerá témata pro nastavení a funkce programu. Celkem se jedná o čtyřicet témat, která svým obsahem a v některých případech i obrazovou ilustrací naprosto vyčerpávajícím způsobem uživatele informují o daném problému. Nápovědu jsem připojil k aplikaci tak, že je možné vyvolávat jednotlivá témata i celou nápovědu.

Ověřovacím měřením jsem zjistil, že mnou navržená aplikace funguje zcela správně. Touto automatizací dojde ke značné úspoře času z pohledu uživatele.

Výsledkem této práce je aplikace, pomocí které lze jednoduše a efektivně měřit izolační odpor izolačních materiálů. Aplikace bude sloužit k dalšímu výzkumu a měření v oblasti degradace izolačních materiálů v asynchronních motorech.

Bohužel výsledky z ověřovacího měření vykazují chyby srovnatelné s manuálním měřením. Tyto chyby jsou způsobeny systematickou chybou měření. Tuto systematickou chybu měření způsobuje především ne zcela profesionální přípravek na uchycení vzorků izolací.

Pro dosažení správných výsledků by bylo nejvhodnější zakoupit níže uvedené přístroje, aby byla eliminována již zmiňovaná systematická chyba měření. Jedná se o profesionální přípravek pro umístění vzorků izolací od firmy KEITHLEY, prodáváný pod označením KEITHLEY 8009. Pro snížení chyby měření by bylo nutné kalibrovat ampérmetr pomocí přístroje KEITHLEY 5156. Při využití multimetru KEITHLEY 6517A a jeho vestavěného zdroje by bylo možné přímo měřit VA charakteristiky izolačních materiálů i závislost odporu na napětí těchto materiálů.

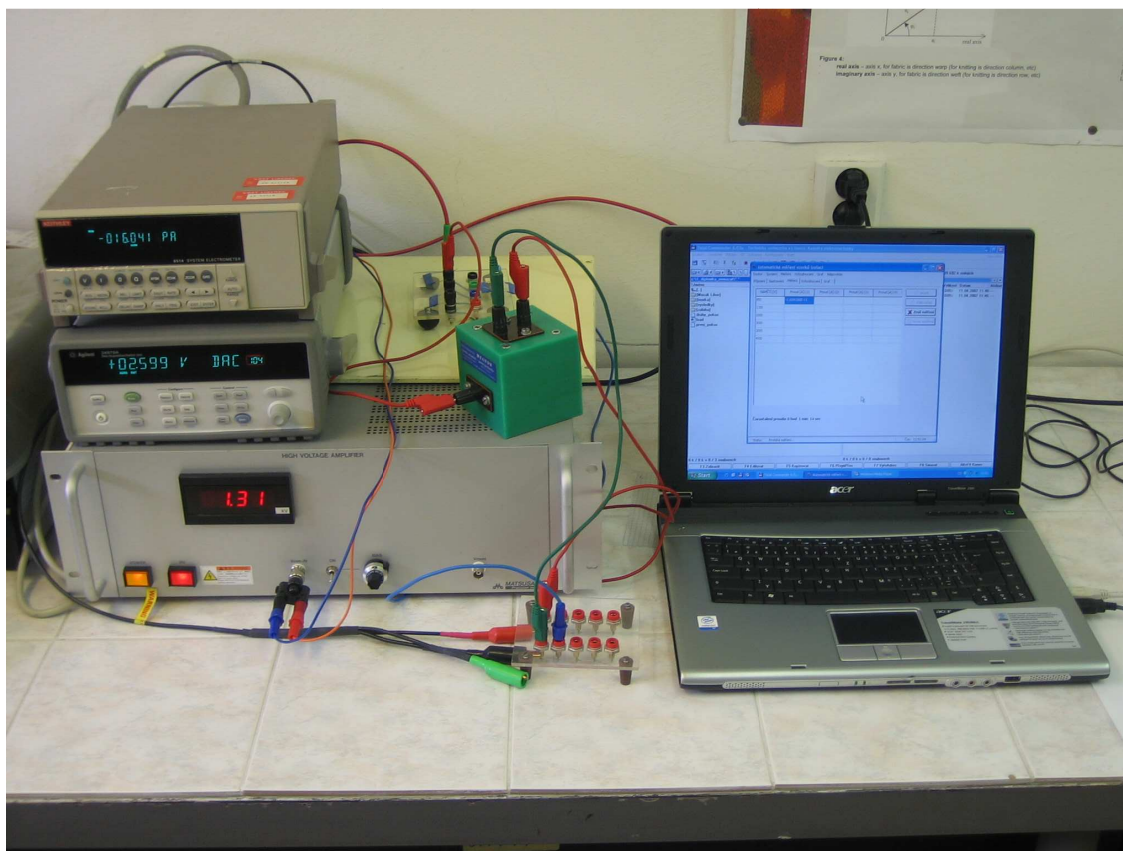
V případě takto vybaveného pracoviště by bylo možné měřit vzorky izolací při napětí do 1000 V, kdy je závislost odporu na napětí lineární (jak je uvedeno v literatuře [1]).



## Literatura

- [1] Artbauer J., Šedovič J., Adamec A. Izolanty a izolácie. 1. vydání. Bratislava: Vydavateľství ALFA, 1969.
- [2] Mušák L. Měření izolačního odporu drážkové izolace. Ročníkový projekt. Liberec, 2006. 51s.
- [3] Haasz V., Roztočil J., Novák J. Číslicové měřicí systémy. 2. vydání. Praha: Vydavatelství ČVUT, 2000. 315 s.
- [4] Cyber Research, GPIB Cables. [online], [cit. 2. 5. 2007]. Dostupné na [www: http://www.cyberresearch.com/store/cables-wiring/pc-computer-cables/ieee-488-gpib-cables/IEEE\\_10-25\\_3023.2.htm](http://www.cyberresearch.com/store/cables-wiring/pc-computer-cables/ieee-488-gpib-cables/IEEE_10-25_3023.2.htm)
- [5] Agilent I/O hardware for PC to instrument connection. [online], [cit. 12. 1. 2007]. Dostupné na [www: http://cfa-www.harvard.edu/~thunter/datasheets/5989-1889EN.pdf](http://cfa-www.harvard.edu/~thunter/datasheets/5989-1889EN.pdf)
- [6] GPIB-PCMCIA Controleur [online], [cit. 19. 12. 2006]. Dostupné na [www: http://www.acquisys.fr/pcmcia\\_gpib.htm](http://www.acquisys.fr/pcmcia_gpib.htm)
- [7] Balíček komponent. [online], [cit. 21.4. 2007]. Dostupné na [www: http://www.lohninger.com/chartpack.html](http://www.lohninger.com/chartpack.html)
- [8] Microsoft Help Workshop [online], [cit. 15.3. 2007]. Dostupné na [www: http://www.margino.com/download](http://www.margino.com/download)
- [9] Matoušek D. C++ Builder 4.0, 5.0 a 6.0, vývojové prostředí - 1. díl. Praha: Vydavatelství BEN, 2004. 688 s. ISBN 80-7300-064-4
- [10] HTML Help Workshop [online], [cit. 15.3. 2007]. Dostupné na [www: http://www.microsoft.com](http://www.microsoft.com)
- [11] Tvorba HTLM helpu – Tvorba obsahu. [online], [cit. 12.3. 2007]. Dostupné na [www: http://www.bozena.net/serialy/chm/chm6.htm](http://www.bozena.net/serialy/chm/chm6.htm)

## PŘÍLOHA A Fotodokumentace



Obrázek pracoviště pro měření izolačního odporu



Ukázka vzorku drážkové izolace

## PŘÍLOHA B Tabulky naměřených hodnot

Zde jsou uvedeny tabulky z manuálního (vlevo) a automatického měření (vpravo).

Hodnoty bez tepelného zatížení:

U[V]	I[pA]
860 ± 6	87 ± 9
1275 ± 8	104 ± 7
2018 ± 8	114 ± 6
3021 ± 8	126 ± 7
4066 ± 11	149 ± 9
4863 ± 7	165 ± 11

U[V]	I[pA]
850 ± 3	58 ± 1
1300 ± 4	62 ± 3
2000 ± 6	69 ± 3
3000 ± 8	71 ± 2
4000 ± 11	75 ± 4
4900 ± 13	75 ± 1

Hodnoty po 8 hodinách tepelného zatížení při 175°C:

U[V]	I[pA]
860 ± 9	119 ± 4
1277 ± 3	122 ± 3
2026 ± 3	128 ± 4
3020 ± 9	132 ± 4
4044 ± 13	136 ± 4
4866 ± 6	138 ± 4

U[V]	I[pA]
850 ± 3	62 ± 3
1300 ± 4	72 ± 4
2000 ± 6	73 ± 4
3000 ± 8	80 ± 2
4000 ± 11	72 ± 3
4900 ± 13	91 ± 3

Hodnoty po 16 hodinách tepelného zatížení při 175°C:

U[V]	I[pA]
876 ± 7	126 ± 4
1271 ± 3	130 ± 3
2013 ± 6	134 ± 4
3027 ± 6	142 ± 2
4019 ± 6	146 ± 2
4865 ± 2	150 ± 4

U[V]	I[pA]
850 ± 3	63 ± 6
1300 ± 4	73 ± 5
2000 ± 6	85 ± 5
3000 ± 8	88 ± 6
4000 ± 11	94 ± 9
4900 ± 13	95 ± 8

Hodnoty po 24 hodinách tepelného zatížení při 175°C:

U[V]	I[pA]
874 ± 7	104 ± 5
1258 ± 4	119 ± 1
2033 ± 5	129 ± 2
3022 ± 3	134 ± 2
4028 ± 4	139 ± 2
4871 ± 6	140 ± 2

U[V]	I[pA]
850 ± 3	65 ± 6
1300 ± 4	71 ± 3
2000 ± 6	77 ± 2
3000 ± 8	82 ± 4
4000 ± 11	85 ± 4
4900 ± 13	94 ± 7

Hodnoty po 32 hodinách tepelného zatížení při 175°C:

U[V]	I[pA]
866 ± 5	99 ± 2
1271 ± 4	107 ± 3
2033 ± 1	115 ± 3
3024 ± 5	122 ± 3
4034 ± 3	125 ± 3
4875 ± 4	129 ± 2

U[V]	I[pA]
850 ± 3	70 ± 5
1300 ± 4	73 ± 5
2000 ± 6	73 ± 7
3000 ± 8	85 ± 8
4000 ± 11	86 ± 6
4900 ± 13	86 ± 6

Hodnoty po 40 hodinách tepelného zatížení při 175°C:

U[V]	I[pA]
873 ± 4	90 ± 2
1273 ± 5	101 ± 1
2025 ± 4	117 ± 2
3028 ± 5	125 ± 3
4022 ± 7	128 ± 2
4880 ± 6	131 ± 1

U[V]	I[pA]
850 ± 3	40 ± 2
1300 ± 4	55 ± 2
2000 ± 6	63 ± 2
3000 ± 8	72 ± 2
4000 ± 11	77 ± 2
4900 ± 13	80 ± 2

Hodnoty po 48 hodinách tepelného zatížení při 175°C:

U[V]	I[pA]
878 ± 6	91 ± 3
1265 ± 3	97 ± 2
2026 ± 4	103 ± 2
3029 ± 7	109 ± 3
4011 ± 4	114 ± 3
4866 ± 4	115 ± 3

U[V]	I[pA]
850 ± 3	46 ± 5
1300 ± 4	35 ± 8
2000 ± 6	54 ± 4
3000 ± 8	54 ± 6
4000 ± 11	64 ± 4
4900 ± 13	65 ± 9

Hodnoty po 56 hodinách tepelného zatížení při 190°C:

U[V]	I[pA]
869 ± 5	95 ± 4
1280 ± 3	101 ± 3
2024 ± 4	109 ± 3
3030 ± 6	116 ± 2
4015 ± 4	119 ± 2
4874 ± 7	93 ± 8

U[V]	I[pA]
850 ± 3	54 ± 6
1300 ± 4	60 ± 8
2000 ± 6	68 ± 6
3000 ± 8	70 ± 4
4000 ± 11	73 ± 6
4900 ± 13	71 ± 6

Hodnoty po 64 hodinách tepelného zatížení při 190°C:

U[V]	I[pA]
880 ± 3	80 ± 2
1275 ± 7	88 ± 1
2028 ± 6	104 ± 5
3039 ± 2	111 ± 6
4015 ± 2	111 ± 6
4872 ± 5	114 ± 5

U[V]	I[pA]
850 ± 3	60 ± 2
1300 ± 4	57 ± 4
2000 ± 6	65 ± 2
3000 ± 8	67 ± 2
4000 ± 11	75 ± 7
4900 ± 13	75 ± 5

Hodnoty po 72 hodinách tepelného zatížení při 190°C:

U[V]	I[pA]
878 ± 5	74 ± 2
1277 ± 8	84 ± 2
2033 ± 4	98 ± 2
3025 ± 7	104 ± 3
4024 ± 3	105 ± 4
4873 ± 4	106 ± 2

U[V]	I[pA]
850 ± 3	62 ± 2
1300 ± 4	63 ± 2
2000 ± 6	68 ± 2
3000 ± 8	72 ± 3
4000 ± 11	72 ± 1
4900 ± 13	76 ± 2

Hodnoty po 80 hodinách tepelného zatížení při 190°C:

U[V]	I[pA]
878 ± 3	28 ± 6
1279 ± 8	31 ± 5
2033 ± 5	36 ± 5
3030 ± 3	42 ± 4
4031 ± 4	47 ± 3
4867 ± 4	58 ± 2

U[V]	I[pA]
850 ± 3	56 ± 5
1300 ± 4	57 ± 6
2000 ± 6	55 ± 6
3000 ± 8	68 ± 4
4000 ± 11	69 ± 4
4900 ± 13	77 ± 5

**PŘÍLOHA C    Tabulka použitých přístrojů**

<i><b>přístroj</b></i>	<i><b>výrobce</b></i>	<i><b>typ</b></i>	<i><b>inv. číslo</b></i>
Ampérmetr	KEITHLEY	6514	ZP 54118
Zesilovač napětí	MATSUSADA	HEOPS-5B6-LCS	54330
Měřicí stanice	AGILENT	34970A	–
Zdroj napětí (zásuvná karta)	AGILENT	34907A	DM 400080
Kabel s rozhraním USB/GPIB	AGILENT	82357A	DM 601560

**PŘÍLOHA D    Obsah CD**